QGAE: an End-to-end Answer-Agnostic Question Generation Model for Generating Question-Answer Pairs

Linfeng Li¹, Licheng Zhang², Chiwei Zhu¹, and Zhendong Mao¹

¹School of Cyber Science and Technology, University of Science and Technology of China, No.443 Huangshan Road, Hefei, 230026, Anhui; ²School of Information Science and Technology, University of Science and Technology of China, No.443 Huangshan Road, Hefei, 230026, Anhui

Correspondence: Zhendong Mao, E-mail: zdmao@ustc.edu.cn

© 2023 The Author(s). This is an open access article under the CC BY-NC-ND 4.0 license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Public summary

- We are the first to propose a new end-to-end model using PLMs, which is called QGAE for answer-agnostic question generation.
- The QGAE model generates question-answer pairs from unannotated texts without requiring any additional information.
- Our model achieves state-of-the-art performance in generating high-quality question-answer pairs, outperforming existing methods by a significant margin.

Citation: Li L F, Zhang L C, Zhu C W, et al. QGAE: an End-to-end Answer-Agnostic Question Generation Model for Generating Question-Answer Pairs. JUSTC, 2023, 53(0). DOI: 10.52396/JUSTC-2023-0002
QGAE: an End-to-end Answer-Agnostic Question Generation Model for Generating Question-Answer Pairs

Lin Feng Li¹, Licheng Zhang², Chiwei Zhu¹, and Zhendong Mao¹

¹School of Cyber Science and Technology, University of Science and Technology of China, No.443 Huangshan Road, Hefei, 230026, Anhui; ²School of Information Science and Technology, University of Science and Technology of China, No.443 Huangshan Road, Hefei, 230026, Anhui

Correspondence: Zhendong Mao, E-mail: zdmao@ustc.edu.cn

© 2023 The Author(s). This is an open access article under the CC BY-NC-ND 4.0 license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Abstract: Question generation aims to generate meaningful and fluent questions, which can address the lack of question-answer type annotated corpus by augmenting the available data. Using unannotated text with optional answers as input contents, question generation can be divided into two types based on whether answers are provided: answer-aware and answer-agnostic. While generating questions with providing answers is challenging, generating high-quality questions without providing answers is even more difficult, for both humans and machines. In order to address this issue, we propose a novel end-to-end model called QGAE, which is able to transform answer-agnostic question generation into answer-aware question generation by directly extracting candidate answers. This approach effectively utilizes unlabeled data for generating high-quality question-answer pairs, and its end-to-end design makes it more convenient compared to a multi-stage method that requires at least two pre-trained models. Moreover, our model achieves better average scores and greater diversity. Our experiments show that QGAE achieves significant improvements in generating question-answer pairs, making it a promising approach for question generation.

Keywords: question generation; natural language processing; answer-agnostic question generation; answer extraction

1 Introduction

Question generation (QG) is defined as the task of generating fluent, meaningful questions automatically from texts with optional answers, so it can be mainly divided into two streams: answer-aware QG that requires answers and answer-agnostic QG that don't. QG is the reverse task of question answering (QA) which is a long-standing and valuable task helping computers achieve machine reading comprehension (MRC), dating back to the 1960s. As with many other supervised learning tasks, QA will also encounter the lack of annotated data in spite of the fact that annotated data sometimes make the most essential part of the whole work.

QG is a popular choice to make data augmentation for QA to alleviate insufficient labeled data. With the continuous development of Internet technology, it is becoming easier and easier to obtain valuable data from the Internet. But question-answer pairs (as shown in Table 1) are still such expensive corpora that typically require manual annotation by crowd-sourcing before being used for supervised learning on QA and QG tasks. To alleviate the high-cost problem of generating question-answer pairs, it’s natural to consider answer-agnostic QG, since its only input is raw text.

Though labeled answers are not necessary, answer-agnostic QG is still facing a great challenge. Most previous works focus on providing additional information to their models by leveraging NER to get extra linguistic features, adding answer position features, using knowledge graphs, and some other methods to improve the generation effect. These methods effectively improve the fluency and accuracy of generated texts, but answer-agnostic QG still performs worse than answer-aware QG. This inspired us that answer-aware QG may play an irreplaceable role, and changing answer-agnostic QG to answer-aware QG is a good choice. Apart from this, there is still an obstacle in generating question-answer pairs that answer-agnostic QG can’t generate answers. To address this issue, researchers often add an additional measure for question-answer pairs generation: answer extraction. Compared with generating an answer, extracting an exact span in the context is much simple.

Extracting candidate answers explicitly not only will resolve the demand for lack of answers but also can change answer-agnostic QG to answer-aware QG. As shown in Figure 1, some works like RGF proposed a multi-stage pipeline method to handle the problem. A multi-stage pipeline method is often designed in complexity, including several parts, and each part may need different inputs. Some early RNN-based works optimized pipeline methods in an end-to-end way, which makes the overall structure lighter and faster.

Though pre-trained language models (PLMs) have occupied dominance in both natural language generation and understanding, there is still no end-to-end work using pre-trained models to generate question-answer pairs. We are sure there
is enough potential for PLMs to achieve the task. In this study, we are motivated by the weak performance of answer-agnostic QG compared to answer-aware QG, inspired by the combination of QG and AE tasks, trying to propose an answer-agnostic question generation model called Question Generation with Answer Extractor (QGAE) to alleviate the high demand for large-scale QA pairs. QGAE is a multi-task model that requires only raw texts as input and can achieve the dual task: answer extraction as well as question generation. We design our model basing the PLM model BART, which has dual encoders and a decoder to generate questions and extract answers in parallel. In our study, question generation is the main task, which is the most challenging part just like all other generation tasks for generated texts' high syntactic diversity and semantic substitutability, so we pay more attention and assign a higher weight to the corresponding module. Therefore answer extraction is considered an auxiliary task. The design not only makes it feasible to turn answer-agnostic question generation into answer-aware question generation but also enables the model to be considered capable of generating question-answer pairs. The contributions of this paper are summarized as follows:

1. We are the first to propose a new end-to-end model using PLMs, which is called QGAE for answer-agnostic question generation.

2. The QGAE model generates question-answer pairs from unannotated texts without requiring any additional information.

3. Our model achieves state-of-the-art performance in generating high-quality question-answer pairs, outperforming existing methods by a significant margin.

The paper is organized as follows. In Section 2, we review the related works of question generation and answer extraction. In Section 3, we formulate the QG task and AE task. In Section 4, we describe each module of our QGAE model. In Section 5, we introduce our experiment in detail. In the last Section 6, we conclude this work and give a detailed analysis.

2 Related Works

2.1 Question Generation

The QG field was devoted great interest by researchers for its great potential benefits, therefore, it has made great progress in such application scenarios as data augmentation, chatbots, machine reading comprehension, and intelligent tutors.

In neural model age, Du et al. proposed the first neural QG model focused on answer-agnostic QG, they investigated the effect of encoding sentence- vs. paragraph-level information by using an attention-based model and found by the size of input.
of input text increases, the evaluation score of output decreases. To deal with the rare or unknown word problem, Gulcehre et al. [20] proposed copy mechanism which is first used in the neural machine translation [21] to solve the out-of-vocabulary (OOV) problem, this mechanism was absorbed in the QG task and widely used. Following the old experience of rule-based QG [22], Wu et al. [23] suggested two new strategies to deal with this task: question type prediction and copy loss mechanism. Du et al. [24] make answer extraction and question generation a combination in an LSTM [25] model including an answer feature embedding, denoting answer span with the usual BIO tagging scheme [26].

When it goes to Transformer-based [28] pre-trained language models (PLMs) era, compared to auto-encoder models, auto-regressive [29] models are widely picked as baselines for the QG task. Laban et al. [30] fine-tuned a GPT2 [31] as the base part of a question-driven news chatbot. Wang et al. [32] leveraged BART to propose QAGS (Question Answering and Generation for Summarization) to evaluate automatic summarization. Bhambhoria et al. [33] leveraged T5 [34] to generate QA pairs for COVID-19 literature. Paranjape et al. [35] develop a Retrieve-Generate-Filter (RGF) technique to create counterfactual evaluation and training data with minimal human supervision, which is a multi-stage job.

Traditional works above have motivated us that it's a considerable method to explicitly infer the candidate answer to transform the answer-agnostic QG into the answer-aware QG. Meanwhile, PLMs with fine-tuning achieved SOTA in lots of NLP fields, becoming benchmarks hard to bypass. In a multi-stage work, researchers will choose different PLMs for different stages in question-answer pairs generation, it's effective but heavy. There's still no end-to-end work to handle the whole task. So we combine answer extraction and question generation using PLMs is what we decide to do, and proposed an end-to-end model extracting answers and generating questions in parallel.

2.2 Answer Extraction

Information extraction (IE) is basically defined as the task of turning the unstructured information expressed in natural language text into a structured 3-tuples representation as (NE1; R; NE2). So answer extraction can be seen as a subfield of IE, expecting to pick the most valuable phrase from tuples, no matter is a named entity, a relation, or their combination: an event. There are lots of IE systems proposed for open-domain. Yahya et al. [36] describe ReNoun, an open information extraction system that complements previous efforts which rely on big knowledge bases by focusing on nominal attributes and on the long tail. Del Corro and Gemulla [37] propose ClausIE, a novel, clause-based approach to open information extraction, which extracts relations and their arguments from natural language text. Also, some rule-based systems using man-made extraction rules are proposed, including verb-based [38], semantic role labeling [39], dependency parse trees [40], etc.

In the era of pre-trained models, auto-encoder [41] models like BERT [42] make great progress in natural language for understanding (NLU) tasks. BERT achieves SOTA in the GLUE [43] score which is a multi-task benchmark including named entities recognition. It's a declaration that large PLMs are blossoming in the IE field, and is going to take the place of traditional methods.

3 Task Definition

Answer-agnostic Question Generation

Answer-agnostic Question Generation aims to generate fluent, meaningful questions \( Q = \{q_1, q_2, \ldots, q_n\} \) from unlabeled input context \( C = \{c_1, c_2, \ldots, c_m\} \) without a specific answer. Suppose the length of the question sequence is \( n \) while the length of the context sequence is \( m \). During training, this task aims to maximize the conditional probability of \( Q \). All relevant parameters in the model, as denoted by \( \theta \):

\[
p(Q|C; \theta) = \prod_{i=1}^{n} p(q_i|C, q_{<i}; \theta)
\]  

where the probability of each \( q_i \) is predicted based on all the words generated previously (i.e., \( q_{<i} \)), and input sentence \( C \).

In our work, we split traditional Answer-agnostic Question Generation into 2 sub-tasks: Answer Extraction and Answer-aware Question Generation as early works do.

Answer Extraction

Answer Extraction supposes there is at least one question-worthy candidate answer in the input context \( C = \{c_1, c_2, \ldots, c_n\} \), and then returns its answer \( A = \{a_1, a_2, \ldots, a_n\} \), where \( A \)'s span is limited by \( C \), therefore, \( 1 \leq i \leq j \leq m \).

Answer-aware Question Generation

Answer-aware Question Generation is similar to Answer-agnostic Question Generation while it provides an additional answer \( A = \{a_1, a_2, \ldots, a_n\} \), \( l \) is the length of Answer:

\[
p(Q|C, A; \theta) = \prod_{i=1}^{n} p(q_i|C, A, q_{<i}; \theta)
\]  

4 Model

4.1 Foundation Model

We choose BART (Bidirectional and Auto-Regressive Transformer) as our foundation model. BART is a sequence-to-sequence model that uses a standard Transformer-based encoder-decoder architecture, inheriting its encoder from BERT's bidirectional encoder and its decoder from GPT's left-to-right decoder, and is particularly effective for text generation as well as reading comprehension tasks. One limitation of BART is that it cannot simultaneously perform NLU and NLG tasks. It excels at tasks such as text generation and reading comprehension individually, but integrating these tasks in a single model remains a challenge. However, with its strong foundation, we believe that BART has the potential to be further improved to handle such tasks effectively.

4.2 QGAE

Question Generation with Answer Extractor (QGAE) is a sequence-to-sequence model as shown in Figure 2 which mainly adopts BART's architecture while adding an additional encoder, so there are two encoders and a decoder. The model first extracts the phrase with high probability as \( A \), then rebuild input \( C \) to \( A, C \). The model will return the rebuild in-
put $A, C$ as well as $Q$.

4.2.1 Answer Extractor Encoder

Answer Extractor Encoder is the first encoder inherited from BART similar to BERT and is used to understand the input context and extract the most valuable phrase. We leverage this encoder by appending an extra linear as a classifier to predict the high probability answer span position. Because BART only supports at most a pair of sequences as input, we choose the highest score answer of all predictions as the candidate answer. This module will focus on the first task answer extraction (AE).

We select cross entropy to calculate the loss of the AE task. $K$ is the number of classes. In this task, class $K$ is the position of the input paragraph span in the range $[0, m - 1]$. $m$ is the input context length. $x_k$ indicates that the $i$-th sample is the $k$-th category. $p$ is the probability distribution of annotated data while $q$ is the probability distribution of prediction data:

$$H(p, q) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} p(x_k) \cdot \log(q(x_k))$$ (3)

Concretely, we put the specific answer into Eq.3, and the equation can be changed as:

$$L_{AE} = \ell(\bar{a}, a) = \frac{1}{N} \sum_{i=1}^{N} L_i$$ (4)

$$L_i = -\sum_{j=1}^{K} t_{i,j} \cdot \log \frac{e^{\theta_{i,j}}}{\sum_{k=1}^{K} e^{\theta_{i,k}}}$$ (5)

where $a$ is the labeled answer span as ground-truth, $\bar{a}$ is the target candidate answer span, and $N$ is the data size. $t_{i,j}$ indicates that the true label of the $i$-th answer is the $k$-th category, which can only take 0 or 1.

4.2.2 Question Generation Encoder-Decoder

Question Generation Encoder-Decoder is mainly derived from BART but adds a unique function leveraging the candidate answer extracted from the first encoder to rebuild input $\langle s \rangle C \langle / s \rangle$ to traditional QG inputs as $\langle s \rangle A \langle / s \rangle C \langle / s \rangle$. Then the module will use rebuilt input to generate text as BART does. This module will focus on the second task question generation (QG).

The loss of the QG task is also cross entropy with the only difference being that we use the labeled questions $\bar{q}$ as ground-truth and prediction questions $q$, and class $K$ is the vocabulary size of the model:

$$L_{QG} = \ell(\bar{q}, q) = \frac{1}{N} \sum_{i=1}^{N} L_i$$ (6)

$$L_i = -\sum_{j=1}^{K} t_{i,j} \cdot \log \frac{e^{\theta_{i,j}}}{\sum_{k=1}^{K} e^{\theta_{i,k}}}$$ (7)

4.2.3 QGAE Loss

QGAE Loss is the loss of the multi-task model, in this work, it’s the sum of answer extraction loss and question generation loss:

$$L = \alpha L_{AE} + (1 - \alpha) L_{QG}$$ (8)

where $\alpha$ is the weight of the AE task as a hyper-parameter.
5 Experiments

5.1 Dataset

Stanford Question Answering Dataset (SQuAD) is the most famous reading comprehension dataset for reversible tasks: question answering and question generation. As the table 2 shows, it has 2 versions, SQuAD1.1 and SQuAD2.0, consisting of questions posed by crowd-workers on a set of Wikipedia articles. Each article has several paragraphs, and each paragraph is asked a set of questions and provided answers, where the answer to every question is a segment of text, or span, from the corresponding reading passage. In SQuAD2.0 because of a percentage of unanswerable questions being added to the dataset, some answers may be null.

5.2 Experiments Settings

We implement our models in HuggingFace\[47\] architecture and fine-tune the model with V100 32GB GPUs. We first fine-tune BART-base on SQuAD2.0 for 2 epochs to get checkpoint BART-base-SQuAD2.0-2epoch (BbS2). Then we use BbS2 to initialize our QGAE model, more specifically, QGAE’s dual encoder is initialized by the BbS2’s encoder twice and some linear layers which don’t exist in BbS2 but in QGAE will be initialized randomly. We set the batch size 20, epoch 3, learning rate 0.00002, dropout 0.2, beam search size 10, max input length 1024, max question size 20, and min question size 3. We perform gradient descent by Adam optimizer\[49\]. The coefficient $\alpha$ of task 1 answer extraction is 0.3 while the coefficient of the question generation task is 0.7.

5.3 Evaluation

We report the evaluation results with four metrics: BLEU, METEOR, ROUGE-L, and Exact Match (EM).

BLEU is an algorithm first for evaluating machine-translated text from one natural language to another, later adopted by the text generation task. BLEU compares n-gram words appearing in candidates and references and punishes too-short sentences with a brevity penalty.

ROUGE is a set of metrics including ROUGE-N, ROUGE-L, ROUGE-W, etc. In this work, we mainly choose ROUGE-L which is the Longest Common Sub-sequence (LCS) based statistic. LCS takes into account sentence-level structure similarity naturally and identifies the longest co-occurring in sequence n-grams automatically.

METEOR is also a metric based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision.

Exact Match measures the percentage of predictions that match any one of the ground truth answers exactly.

As each paragraph in the SQuAD dataset may have several question-answer pairs, we use paragraphs as input and compare outputs with a group of question-answer pairs and choose the highest score with BLEU-4 as the main indicator.

6 Results and Discussion

6.1 Results

In Table 3, we compare our proposed end-to-end QGAE model with 3 other types of earlier works: standalone answer extraction task, standalone answer-agnostic question generation, and multi-stage QA-pair generation pipeline. All the data used in the experiments has been replicated from the following paper.

(1) Standalone answer extraction

- KPE: Key Phrase Extraction (KPE)\[43\] is a part of a neural question-answer pair generation system. It has two approaches: KPE-class and KPE-Gen.

(2) Standalone answer-agnostic question generation

- Attention LSTM: Attention LSTM is proposed by Du\[45\] and is the first work focus on answer-agnostic QG.

- Self-Attention Transformers: Self-Attention Transformers\[44\] explore how Transformers can be adapted to the task of Neural Question Generation without constraining the model to focus on a specific answer passage.

- Question-driven LSTM: Question-driven LSTM\[46\] proposed two new strategies question type prediction and copy loss mechanism to deal with the task.

(3) Multi-stage QA-pair generation pipeline

- MCF: Wang et al.\[47\] proposed a multi-stage framework that can extract question-worthy phrases and improve the performance of question generation. We chose this framework as the baseline for the specific task of generating QA pairs and used it to evaluate the performance.

6.2 Discussion

The performance shows that our end-to-end QGAE model not
Table 3. Comparison of method performance in major metrics (including QG metrics and AE metric) on the SQuAD dataset. These methods are divided into four types according to their primary research fields. The first two classifications focus on their own independent fields, while the latter two classifications can accomplish these two tasks at the same time.

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>QG metrics</th>
<th>AE metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BLEU-1</td>
<td>BLEU-2</td>
</tr>
<tr>
<td>Answer Extraction</td>
<td>KPE-Class</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>KPE-Gen</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Answer-agnostic Question Generation</td>
<td>Attention LSTM</td>
<td>43.09</td>
<td>25.96</td>
</tr>
<tr>
<td></td>
<td>Self-Attention Transformer</td>
<td>43.33</td>
<td>26.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45.08</td>
<td>27.98</td>
</tr>
<tr>
<td>Multi-stage(Baseline)</td>
<td>MCF</td>
<td>45.70</td>
<td>25.87</td>
</tr>
<tr>
<td>End-to-end</td>
<td>QGAE</td>
<td>68.32</td>
<td>45.41</td>
</tr>
</tbody>
</table>

only achieves SOTA in the answer extraction task but also makes a great improvement in the answer-agnostic question generation compared with the traditional encoder-decoder architecture. Even if multi-stage work MCF has a much more complex workflow, it has a weaker comprehension performance than our work. What's more? QGAE is lighter, more convenient, and more portable since it only requires fine-tuning of one pre-trained model, whereas multi-stage methods need at least two models for stage AE and QG.

Although great progress has been made in the EM score, reaching 53.82%, there is still a lot of room for improvement in extraction accuracy. Our model may extract candidate answers which are not ground truth but also meaningful, while extraction accuracy is judged and limited by the labeled data. Specifically, the range of candidate answers is very wide, ranging from named entities to relationships, to events. However, only a little percentage of key phrases are included in the training dataset while others are out of range. Candidate answers beyond the confines of the dataset may make the later question generation task in the wrong direction, performing worse when choosing traditional machine-translation evaluation indicators. Despite all this, prediction sentences not in the ground truth are still valuable and reasonable. The high diversity of generated sentences, to a certain extent, is an advantage that will make our model competitive in different scenes for data augmentation.

Therefore it can be concluded: we have expanded our model's function not only to generate questions but also to generate QA-pairs compared to the baseline model and better than any previous work, which proved our model is diverse and efficient.

7 Conclusions

In this paper, our focus is on answer-agnostic question generation, which can be extended to question-answer pair generation. This task can be divided into two sub-tasks: answer extraction and question generation. We proposed an end-to-end model called Question Generation with Answer Extractor (QGAE) using raw text without costing any additional information, which can generate question-answer pairs in parallel. Compared to the multi-stage question-answer generation method, QGAE has several advantages. Firstly, QGAE is able to generate question-answer pairs in parallel, whereas the multi-stage method requires multiple rounds of generation and refinement. Secondly, it is lighter, more convenient, and more portable than multi-stage methods in training, which reduces the complexity of the overall system. Thirdly, our model achieves a better average score and greater diversity. Overall, QGAE is a more efficient and versatile approach to answer-agnostic question generation, with potential applications in various natural language processing tasks.

In further work, we will try to compile more datasets into one ensemble to improve the accuracy of answer extraction. Not only that, we will try to change our main task to information retrieval to optimize our answer extraction, as different weight biases in sub-tasks lead to an imbalance in the model's focus in the two sub-tasks. All in all, this is still pioneering work in pre-trained language models adapting question-answer pairs generation.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for Central Universities under Grants WK3480000010 and WK3480000008.

Conflict of Interest

The authors declare no conflict of interest.

Biographies

Linfeng Li is currently pursuing a master’s degree at the School of Cyber Science and Technology, University of Science and Technology of China.

Licheng Zhang is currently pursuing a doctor’s degree at the School of Information Science and Technology, University of Science and Technology of China.

Chiwei Zhu is currently pursuing a doctor’s degree at the School of Cyberspace Security, University of Science and Technology of China.

Zhendong Mao received his Ph.D. degree in computer application technology from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2014. He is currently a Professor at the School of Cyber Science and Technology, University of Science and Technology of China, Hefei, China. From 2014 to 2018, he was an
References


[29] Pandit S M, Wu S M, Šmits T I. Time series and system analysis


