JUSTC

Article

http://justc.ustc.edu.cn

Received: January 08, 2023; Accepted: April 14, 2023

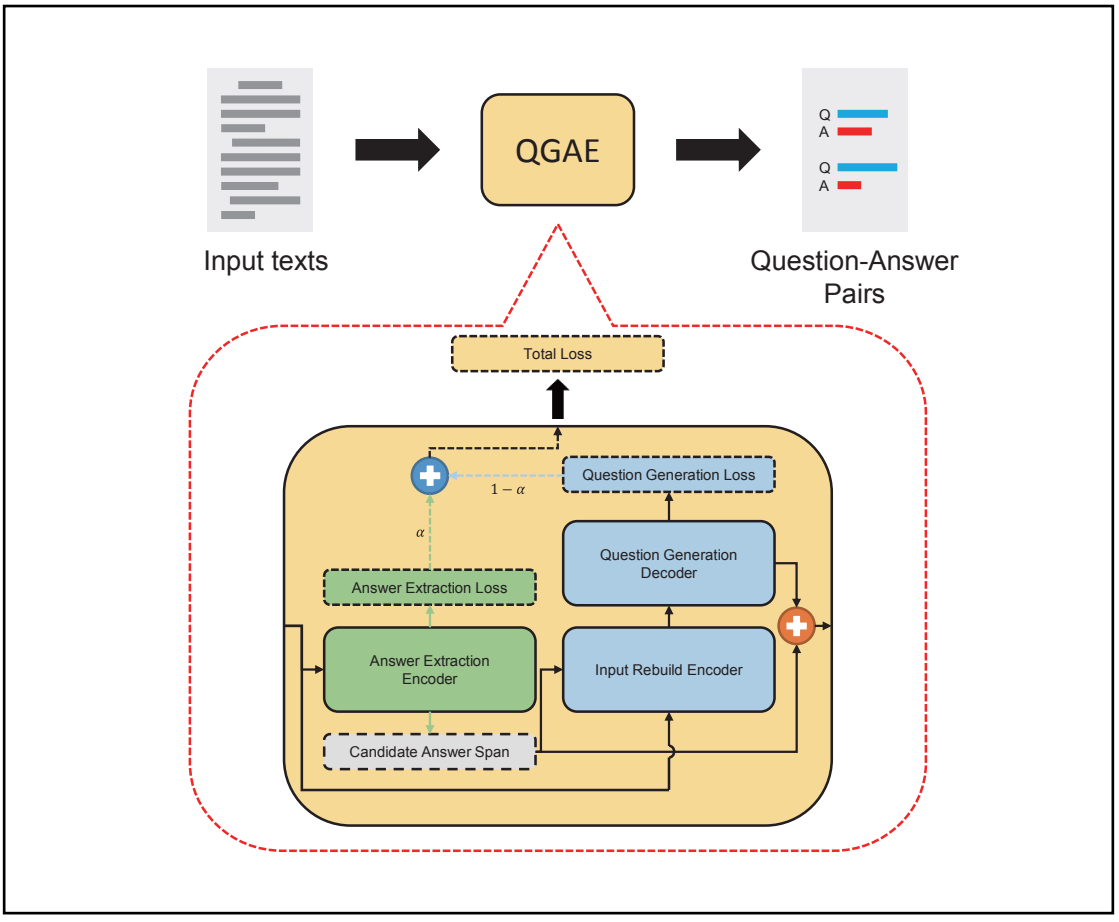
QGAE: an end-to-end answer-agnostic question generation model for generating question-answer pairs

Linfeng Li[1], Licheng Zhang[2], Chiwei Zhu[1], and Zhendong Mao[1] ✉

[1]*School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230027, China;*
[2]*School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China*

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

Graphical abstract



The architecture of the QGAE model.

Public summary

- We propose a new end-to-end question generation model using PLMs for answer-agnostic question generation.
- Our model combines question generation and answer extraction into dual tasks to achieve answer-question pair generation.

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*, **2024**, 54(1): 0102. DOI: 10.52396/JUSTC-2023-0002

JUSTC

Article

http://justc.ustc.edu.cn

Received: January 08, 2023; Accepted: April 14, 2023

QGAE: an end-to-end answer-agnostic question generation model for generating question-answer pairs

Linfeng Li[1], Licheng Zhang[2], Chiwei Zhu[1], and Zhendong Mao[1] ✉

[1]*School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230027, China;*
[2]*School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China*

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

Cite This: *JUSTC*, **2024**, 54(1): 0102 (8pp)

Read Online

Abstract: Question generation aims to generate meaningful and fluent questions, which can address the lack of a 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 by providing answers is challenging, generating high-quality questions without providing answers is even more difficult for both humans and machines. To address this issue, we proposed a novel end-to-end model called question generation with answer extractor (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 than 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: deep learning; natural language processing; answer-agnostic question generation; answer extraction

CLC number: TP391.1 **Document code:** A

1 Introduction

Question generation[1, 2] (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[3] that requires answers, and answer-agnostic QG[4] that does not. QG is the reverse task of question answering (QA), which is a long-standing and valuable task helping computers achieve machine reading comprehension[5, 6], dating back to the 1960s[7]. As with many other supervised learning[8, 9] 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 for data augmentation for QA to alleviate insufficient labeled data. With the continuous development of Internet technology, it is becoming increasingly easier to obtain valuable data from the Internet. However, question-answer pairs (as shown in Table 1) are still such expensive corpora that typically require manual annotation by crowdsourcing before being used for supervised learning on QA and QG tasks. To alleviate the high-cost problem of generating question-answer pairs, it is natural to consider answer-agnostic QG, since its only input is raw text.

Although labeled answers are not necessary, answer-agnostic QG is still facing a great challenge. Most previous works focused on providing additional information to their models by leveraging named entity recognition (NER)[10] to obtain extra linguistic features, adding answer position features[11], using knowledge graphs[12], 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. Thus, 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 pair generation: answer extraction. Compared with generating an answer, extracting an exact span in the context is much simpler.

Explicitly extracting candidate answers will not only resolve the demand for the lack of answers but also can transform answer-agnostic QG into answer-aware QG. As shown in Fig. 1, some works such as RGF[13] (retrieve-generate filter) 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[14–17] 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.

DOI: 10.52396/JUSTC-2023-0002
JUSTC, **2024**, 54(1): 0102

Table 1. A case of QA-pairs generated by our QGAE model: the model accepts unannotated texts as input, extracts the highlighted phrase "Lorentz's law" as an answer, then uses this answer to make question generation.

Input context: Through combining the definition of electric current as the time rate of change of electric charge, a rule of vector multiplication called **Lorentz's law** describes the force on a charge moving in a magnetic field. The connection between electricity and magnetism allows for the description of a unified electromagnetic force that acts on a charge. This force can be written as a sum of the electrostatic force (due to the electric field) and the magnetic force (due to the magnetic field).
Extracted answer: Lorentz's law
Generated question: What describes the force on a charge moving in a magnetic field?

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 tasks: answer extraction and question generation. We design our model based on the PLM model BART[18], 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 similar to 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:

- 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.

The rest of this 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 application scenarios such as data augmentation[19], chatbots[20], machine reading comprehension[21], and intelligent tutors[22].

In the neural model age, Du et al.[4] 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 that

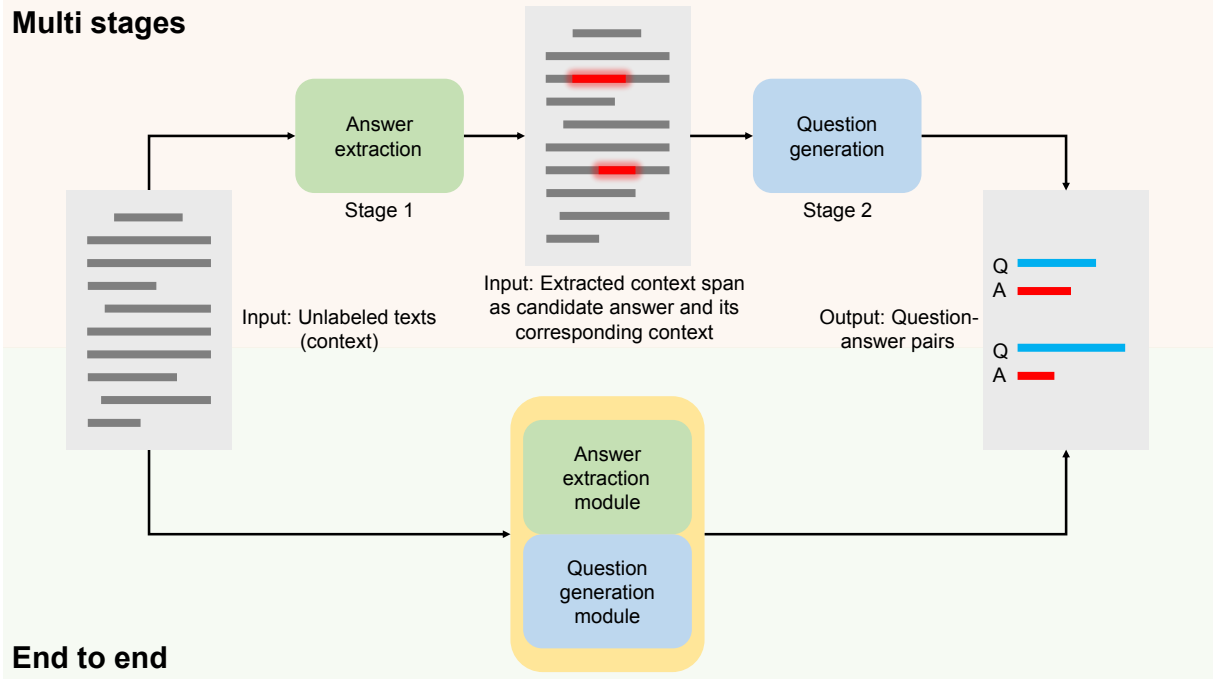


Fig. 1. The difference between multi-stage methods and end-to-end models is that a multi-stage method usually has more than one model in the whole workflow. In every stage, a multi-stage method may need to deal with different inputs and outputs, while on the contrary, an end-to-end model only needs a definite kind of input.

DOI: 10.52396/JUSTC-2023-0002
JUSTC, **2024**, 54(1): 0102

as the size of the input text increased, the evaluation score of the output decreased. To deal with the rare or unknown word problem, Gulcehre et al.[23] proposed a copy mechanism that was first used in the neural machine translation[24] to solve the out-of-vocabulary problem. This mechanism was absorbed in the QG task and widely used. Following the old experience of rule-based QG[25], Wu et al.[26] suggested two new strategies to deal with this task: question type prediction and a copy loss mechanism. Du et al.[15] combined answer extraction and question generation in an LSTM[27] model including answer feature embedding, denoting answer span with the usual BIO tagging scheme[28].

In the transformer-based[29] PLM era, compared to auto-encoder models, auto-regressive[30] models are widely picked as baselines for the QG task. Laban et al.[20] 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.[13] developed a retrieve-generate filter (RGF) technique to create counterfactual evaluation and training data with minimal human supervision, which is a multi-stage job.

The traditional works above have motivated us 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 many NLP fields, becoming benchmarks hard to bypass. In multi-stage work, researchers will choose different PLMs for different stages in question-answer pair generation, which is effective but heavy. There's still no end-to-end work to handle the whole task. Therefore, we combine answer extraction and question generation using PLMs and propose an end-to-end model that extracts answers and generates questions in parallel.

2.2 Answer extraction

Information extraction[35, 36] (IE) is basically defined as the task of turning the unstructured information expressed in natural language text into a structured 3-tuple representation as (NE1; R; NE2). Thus, answer extraction can be seen as a sub-field of IE, expecting to pick the most valuable phrase from tuples, regardless of whether it is a named entity, a relation, or their combination: an event. Many IE systems have been proposed for open domains. Yahya et al.[37] describe ReNoun, an open information extraction system that complements previous efforts that rely on big knowledge bases by focusing on nominal attributes and on the long tail. Del Corro and Gemulla[38] proposed ClausIE, a novel, clause-based approach to open information extraction, which extracts relations and their arguments from natural language text. Additionally, some rule-based systems using man-made extraction rules have been proposed, including verb-based[39], semantic role labeling[40], and dependency parse trees[41].

In the era of pre-trained models, auto-encoder[42] models, such as BERT[43] have made great progress in natural language understanding (NLU) tasks. BERT achieves SOTA in the GLUE[44] score which is a multi-task benchmark including named entity recognition. It is a declaration that large PLMs are blossoming in the IE field and will take the place of traditional methods.

3 Task definition

Answer-agnostic question generation. It aims to generate fluent, meaningful questions $Q = \{q_1, q_2, \cdots, q_n\}$ from unlabeled input context $C = \{c_1, c_2, \cdots, c_m\}$ 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 are denoted by θ:

$$p(Q|C;\theta) = \prod_{t=1}^{n} p(q_t|C, q_{i<t};\theta), \tag{1}$$

where the probability of each q_t is predicted based on all the words generated previously (i.e., $q_{i<t}$), 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 in early works.

Answer extraction. It supposes there is at least one question-worthy candidate answer in the input context $C = \{c_1, c_2, \cdots, c_m\}$ and then returns its answer $A = \{a_i, a_{i+1}, \cdots, a_j\}$, where A's span is limited by C, therefore, $1 \leqslant i \leqslant j \leqslant m$.

Answer-aware question generation. It is similar to answer-agnostic question generation while it provides an additional answer $A = \{a_1, a_2, \cdots, a_l\}$, l is the length of the answer:

$$p(Q|C,A;\theta) = \prod_{t=1}^{n} p(q_t|C, A, q_{i<t};\theta). \tag{2}$$

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 (natural language generation) 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

QGAE is a sequence-to-sequence model as shown in Fig. 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 and rebuilds input C to A, C. The model will return the rebuild input A, C, and 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

DOI: 10.52396/JUSTC-2023-0002
JUSTC, 2024, 54(1): 0102

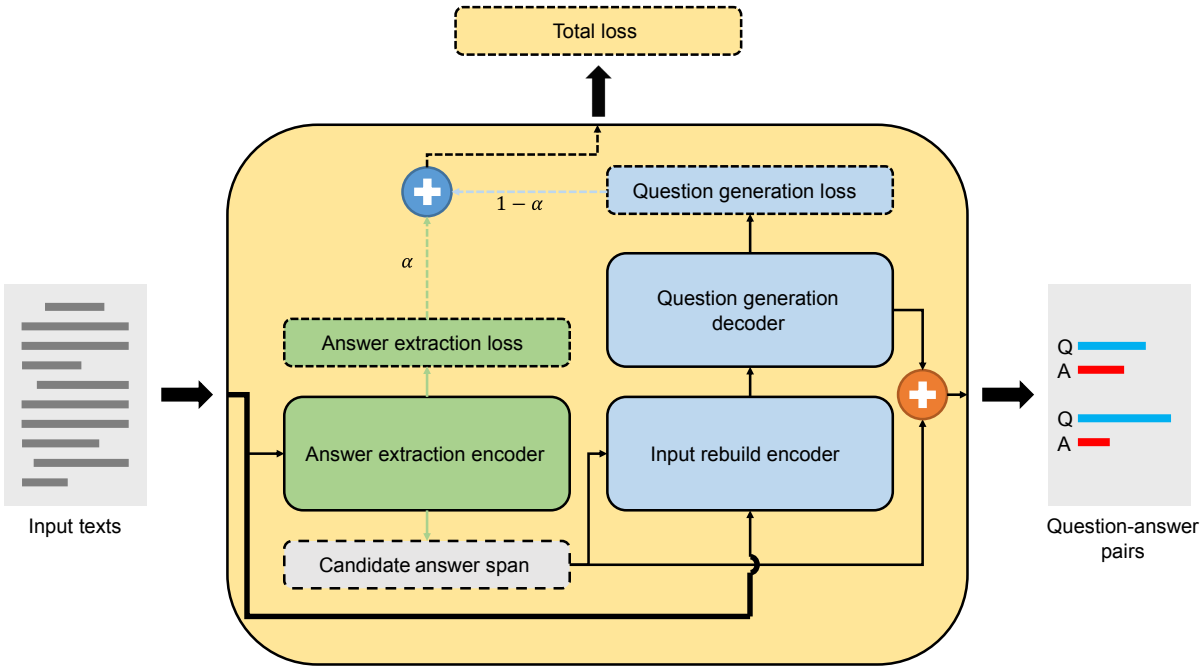


Fig. 2. The architecture of QGAE consists of two encoders and one decoder, which take raw texts as input and generate question-answer pairs.

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 $[0, m-1]$, and m is the input context length. $x_{i,k}$ indicates that the ith sample is the kth category. p is the probability distribution of annotated data while q is the probability distribution of prediction data:

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

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

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

$$L_i = -\sum_{k=1}^{K} t_{i,k} \cdot \log \frac{\mathrm{e}^{\bar{a}_{i,k}}}{\sum_{j=1}^{K} \mathrm{e}^{\bar{a}_{i,j}}}, \tag{5}$$

where a is the labeled answer span as ground-truth, $\bar{a}$ is the target candidate answer span, and N is the data size. $t_{i,k}$ indicates that the true label of the ith answer is the kth 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 $\langle s\rangle A\langle /s\rangle\langle /s\rangle C\langle /s\rangle$. Then, the module uses 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 q as ground-truth and prediction questions $\bar{q}$, and class K is the vocabulary size of the model:

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

$$L_i = -\sum_{k=1}^{K} t_{i,k} \cdot \log \frac{\mathrm{e}^{\bar{q}_{i,k}}}{\sum_{j=1}^{K} \mathrm{e}^{\bar{q}_{i,j}}}. \tag{7}$$

4.2.3 QGAE loss

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

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

where α is the weight of the AE task as a hyper-parameter.

5 Experiments

5.1 Dataset

The 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 two versions, SQuAD1.1[45] and

DOI: 10.52396/JUSTC-2023-0002
JUSTC, **2024**, 54(1): 0102

SQuAD2.0[46], consisting of questions posed by crowdworkers 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 are 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 32 GB GPUs. We first fine-tune BART-base on SQuAD2.0 for 2 epochs to obtain checkpoint BART-base-SQuAD2.0-2 epoch (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 that do not exist in BbS2 but in the QGAE will be initialized randomly. We set the batch size to 20, epoch to 3, learning rate to 0.00002, dropout to 0.2, beam search size to 10, max input length to 1024, max question size to 20, and min question size to 3. We perform gradient descent by the Adam optimizer[48]. The coefficient α 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).

Table 2. Statistics of datasets SQuAD1.1 and SQuAD2.0. No matter in which dataset, an example consists of a context, a question, and an optional answer. The term "negative example" refers to a context passage paired with an unanswerable question, which is intended to help models learn to identify when a question cannot be answered correctly based on the given context.

Dataset	SQuAD1.1	SQuAD2.0
Total		
Number of articles	536	505
Total examples	107702	151054
Train		
Number of articles	442	442
Articles with negatives	0	285
Total examples	87599	130319
Negative examples	0	43498
Development		
Number of articles	48	35
Articles with negatives	0	35
Total examples	10570	11873
Negative examples	0	5945
Test		
Number of articles	46	28
Articles with negatives	0	28
Total examples	9533	8862
Negative examples	0	4332

BLEU. 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. ROUGE is a set of metrics including ROUGE-N, ROUGE-L, and ROUGE-W. 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. METEOR is also a metric based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision.

Exact match. 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 have been replicated from the following paper.

(Ⅰ) Standalone answer extraction

KPE. Key phrase extraction (KPE)[49] is a part of a neural question-answer pair generation system. It has two approaches: KPE-class and KPE-Gen.

(Ⅱ) Standalone answer-agnostic question generation

Attention LSTM. Attention LSTM was proposed by Du et al.[4] and was the first work to focus on answer-agnostic QG.

Self-attention transformers. Self-attention transformers[50] 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[26] proposed two new strategies question type prediction and a copy loss mechanism to address the task.

(Ⅲ) Multi-stage QA-pair generation pipeline

MCF. Wang et al.[51] 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 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, has a weaker comprehensive

DOI: 10.52396/JUSTC-2023-0002
JUSTC, **2024**, 54(1): 0102

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.

Method	Model	QG metrics						AE metric
		BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	EM
Answer extraction	KPE-Class	-	-	-	-	-	-	20.66%
	KPE-Gen	-	-	-	-	-	-	36.50%
Answer-agnostic question generation	Attention LSTM	43.09	25.96	17.50	12.28	16.62	39.75	-
	Self-attention transformer	43.33	26.27	18.32	13.23	-	40.22	-
		45.08	27.98	19.38	13.90	18.12	40.77	-
Multi-stage (Baseline)	MCF	45.70	25.87	16.33	10.56	15.76	38.09	35.77%
End-to-end	**QGAE**	**68.32**	**45.41**	**30.68**	**20.11**	**48.97**	**45.66**	**53.82%**

performance than our work. What is 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 much room for improvement in extraction accuracy. Our model may extract candidate answers that 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 small 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 that 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. First, QGAE is able to generate question-answer pairs in parallel, whereas the multi-stage method requires multiple rounds of generation and refinement. Second, it is lighter, more convenient, and more portable than multi-stage methods in training, which reduces the complexity of the overall system. Third, 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 pair generation.

Acknowledgements

This work was supported by the Fundamental Research Funds for Central Universities (WK3480000010, WK3480000008).

Conflict of interest

The authors declare that they have 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. His research interest is natural language processing.

Zhendong Mao received his Ph.D. degree in Computer Application Technology from the Institute of Computing Technology, Chinese Academy of Sciences (CAS) in 2014. From 2014 to 2018, he was an Assistant Professor at the Institute of Information Engineering, CAS. He is currently a Professor at the School of Cyber Science and Technology, University of Science and Technology of China. His research interests include the fields of computer vision, natural language processing, and cross-modal understanding.

References

[1] Rus V, Cai Z, Graesser A. Question generation: Example of a multi-year evaluation campaign. In: Proceedings of 1st Question Generation Workshop, **2008**.

[2] Rus V, Wyse B, Piwek P, et al. The first question generation shared task evaluation challenge. In: Proceedings of the 6th International Natural Language Generation Conference. New York: ACM, **2010**: 251–257.

[3] Wang B, Wang X, Tao T, et al. Neural question generation with answer pivot. *Proceedings of the AAAI Conference on Artificial*

DOI: 10.52396/JUSTC-2023-0002
JUSTC, **2024**, 54(1): 0102

Intelligence, **2020**, *34*: 9138–9145.
[4] Du X, Shao J, Cardie C. Learning to ask: Neural question generation for reading comprehension. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, **2017**: 1342–1352.
[5] Baradaran R, Ghiasi R, Amirkhani H. A survey on machine reading comprehension systems. *Natural Language Engineering,* **2022**, *28*: 683–732.
[6] Chen D. Neural Reading Comprehension and Beyond. Stanford, California: Stanford University, **2018**.
[7] Green B F Jr, Wolf A K, Chomsky C, et al. Baseball: An automatic question-answerer. In: Papers presented at the May 9–11, 1961, western joint IRE-AIEE-ACM computer conference. New York: ACM Press, **1961**: 219–224.
[8] Cunningham P, Cord M, Delany S J. Supervised learning. In: Cord M, Cunningham P, editors. Machine Learning Techniques for Multimedia. Cognitive Technologies. Berlin, Heidelberg: Springer, **2008**: 21–49.
[9] Liu B. Supervised learning. In: Web Data Mining. Berlin, Heidelberg: Springer, **2011**: 63–132.
[10] Zhang S, Bansal M. Addressing semantic drift in question generation for semi-supervised question answering. 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). Hong Kong, China: Association for Computational Linguistics, **2019**: 2495–2509.
[11] Zhou Q, Yang N, Wei F, et al. Neural question generation from text: A preliminary study. In: National CCF conference on natural language processing and Chinese computing. Cham, Switzerland: Springer, **2018**: 662–671.
[12] Reddy S, Raghu D, Khapra M M, et al. Generating natural language question-answer pairs from a knowledge graph using an RNN-based question generation model. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain: Association for Computational Linguistics. **2017**: 376–385.
[13] Paranjape B, Lamm M, Tenney I. Retrieval-guided counterfactual generation for QA. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Dublin, Ireland: Association for Computational Linguistics, **2022**: 1670–1686.
[14] Du X, Cardie C. Identifying where to focus in reading comprehension for neural question generation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics, **2017**: 2067–2073.
[15] Du X, Cardie C. Harvesting paragraph-level question-answer pairs from Wikipedia. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, **2018**: 1907–1917.
[16] Kumar V, Ramakrishnan G, Li Y F. Putting the horse before the cart: A generator-evaluator framework for question generation from text. In: Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL). Hong Kong, China: Association for Computational Linguistics, **2019**: 812–821.
[17] Nakanishi M, Kobayashi T, Hayashi Y. Towards answer-unaware conversational question generation. In: Proceedings of the 2nd Workshop on Machine Reading for Question Answering. Hong Kong, China: Association for Computational Linguistics, **2019**: 63–71.
[18] Lewis M, Liu Y, Goyal N, et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, **2020**: 7871–7880.
[19] Kumar V, Black A W. ClarQ: A large-scale and diverse dataset for Clarification Question Generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, **2020**: 7296–7301.
[20] Laban P, Canny J, Hearst M A. What's the latest? A question-driven news chatbot. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Online: Association for Computational Linguistics, **2020**: 380–387.
[21] Yuan X, Wang T, Gulcehre C, et al. Machine comprehension by text-to-text neural question generation. In: Proceedings of the 2nd Workshop on Representation Learning for NLP. Vancouver, Canada: Association for Computational Linguistics, **2017**: 15–25.
[22] Yao B, Wang D, Wu T, et al. It is AI's turn to ask humans a question: Question-answer pair generation for children's story books. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Dublin, Ireland: Association for Computational Linguistics, **2022**: 731–744.
[23] Gulcehre C, Ahn S, Nallapati R, et al. Pointing the unknown words. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, **2016**: 140–149.
[24] Kalchbrenner N, Blunsom P. Recurrent continuous translation models. In: 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, USA: Association for Computational Linguistics, **2013**: 1700–1709.
[25] Mostow J, Chen W. Generating instruction automatically for the reading strategy of self-questioning. In: Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling. Brighton, UK: ACM, **2009**: 465–472.
[26] Wu X, Jiang N, Wu Y. A question type driven and copy loss enhanced framework for answer-agnostic neural question generation. In: Proceedings of the Fourth Workshop on Neural Generation and Translation. Online: Association for Computational Linguistics, **2020**: 69–78.
[27] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation,* **1997**, *9*: 1735–1780.
[28] Ramshaw L A, Marcus M P. Text chunking using transformation-based learning. In: Armstrong S, Church K, Isabelle P, editors. Natural Language Processing Using Very Large Corpora. Dordrecht: Springer, **1999**: 157–176.
[29] Vaswani A, Shazeer N, Parmar N, et al. Attention is all You need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM, **2017:** 6000–6010.
[30] Pandit S M, Wu S M, Šmits T I. Time series and system analysis with applications by Sudhakar Madhavrao Pandit and Shien-Ming Wu. *The Journal of the Acoustical Society of America,* **1984**, *75*: 1924–1925.
[31] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners. OpenAI blog, **2019**. https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf. Accessed December 8, 2022.
[32] Wang A, Cho K, Lewis M. Asking and answering questions to evaluate the factual consistency of summaries. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, **2020**: 5008–5020.
[33] Bhambhoria R, Feng L, Sepehr D, et al. A smart system to generate and validate question answer pairs for COVID-19 literature. In: Proceedings of the First Workshop on Scholarly Document Processing. Online: Association for Computational Linguistics,

DOI: 10.52396/JUSTC-2023-0002
JUSTC, **2024**, 54(1): 0102

2020: 20–30.

[34] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research,* **2020**, *21* (1): 5485–5551.

[35] Niklaus C, Cetto M, Freitas A, et al. A survey on open information extraction. In: Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, USA: Association for Computational Linguistics, **2018**: 3866–3878.

[36] Mintz M, Bills S, Snow R, et al. Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2. Suntec, Singapore: Association for Computational Linguistics, **2009**: 1003–1011.

[37] Yahya M, Whang S, Gupta R, et al. ReNoun: Fact extraction for nominal attributes. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, **2014**: 325–335.

[38] Del Corro L, Gemulla R. ClausIE: Clause-based open information extraction. In: Proceedings of the 22nd International Conference on World Wide Web. New York: ACM, **2013**: 355–366.

[39] Fader A, Soderland S, Etzioni O. Identifying relations for open information extraction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. New York: ACM, **2011**: 1535–1545.

[40] Christensen J, Soderland S, Etzioni O, et al. Semantic role labeling for open information extraction. In: Proceedings of the NAACL HLT 2010 first international workshop on formalisms and methodology for learning by reading. Los Angeles, USA: Association for Computational Linguistics, **2010**: 52–60.

[41] Mesquita F, Schmidek J, Barbosa D. Effectiveness and efficiency of open relation extraction. In: 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, USA: Association for Computational Linguistics, **2013**: 447–457.

[42] Dai A M, Le Q V. Semi-supervised sequence learning. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. Cambridge, USA: MIT Press, **2015:** 3079–3087.

[43] Devlin J, Chang M W, Lee K, et al. 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, Volume 1 (Long and Short Papers). Minneapolis, MN, USA: Association for Computational Linguistics, **2019:** 4171–4186.

[44] Wang A, Singh A, Michael J, et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. Brussels, Belgium: Association for Computational Linguistics, **2018**: 353–355.

[45] Rajpurkar P, Zhang J, Lopyrev K, et al. SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, TX, USA: Association for Computational Linguistics, **2016**: 2383–2392.

[46] Rajpurkar P, Jia R, Liang P. Know what You don't know: Unanswerable questions for SQuAD. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Melbourne, Australia: Association for Computational Linguistics, **2018**: 784–789.

[47] Wolf T, Debut L, Sanh V, et al. HuggingFace's transformers: State-of-the-art natural language processing. arXiv: 1910.03771, **2019**.

[48] Kingma D P, Ba J L. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015-Conference Track Proceedings, San Diego, USA: ICLR, **2015**: 7–9.

[49] Willis A, Davis G, Ruan S, et al. Key phrase extraction for generating educational question-answer pairs. In: Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale. New York: ACM, **2019**: 20.

[50] Scialom T, Piwowarski B, Staiano J. Self-attention architectures for answer-agnostic neural question generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, **2019**: 6027–6032.

[51] Wang S, Wei Z, Fan Z, et al. A multi-agent communication framework for question-worthy phrase extraction and question generation. *Proceedings of the AAAI Conference on Artificial Intelligence,* **2019**, *33* (1): 7168–7175.

DOI: 10.52396/JUSTC-2023-0002
JUSTC, **2024**, 54(1): 0102