QA4QG: USING QUESTION ANSWERING TO CONSTRAIN MULTI-HOP QUESTION GENERATION

Dan Su, Peng Xu and Pascale Fung

Department of Electronic and Computer Engineering
Center for Artificial Intelligence Research (CAiRE)
The Hong Kong University of Science and Technology, Clear Water Bay
dsu@connect.ust.hk, pxuab@connect.ust.hk, pascale@ece.ust.hk

ABSTRACT
Multi-hop question generation (MQG) aims to generate complex questions which require reasoning over multiple pieces of information of the input passage. Most existing work on MQG has focused on exploring graph-based networks to equip the traditional Sequence-to-sequence framework with reasoning ability. However, these models do not take full advantage of the constraint between questions and answers. Furthermore, studies on multi-hop question answering (QA) suggest that Transformers can replace the graph structure for multi-hop reasoning. Therefore, in this work, we propose a novel framework, QA4QG, a QA-augmented BART-based framework for MQG. It augments the standard BART model with an additional multi-hop QA module to further constrain the generated question. Our results on the HotpotQA dataset show that QA4QG outperforms all state-of-the-art models, with an increase of 8 BLEU-4 and 8 ROUGE points compared to the best results previously reported. Our work suggests the advantage of introducing pre-trained language models and QA module for the MQG task.

Index Terms— Pre-trained Language Models, Multi-hop Generation, Question Generation, Question Answering

1. INTRODUCTION

Question generation (QG) is the task of automatically generating a question from a given context and an answer. It can be an essential component in education systems [1], or be applied in intelligent virtual assistant systems to make them more proactive. It can also serve as a complementary task to boost QA systems [2].

Most of the previous works on QG focus on generating the SQuAD-style single-hop question, which is relevant to one fact obtainable from a single sentence. Recently, there has been a surge of interest in QG for more complex multi-hop question generation, such as HotpotQA-style questions [3, 4, 5, 6, 7, 8, 9]. This is a more challenging task that requires identifying multiple relevant pieces of information from multiple paragraphs, and reasoning over them to fulfill the generation. Due to the multi-hop nature of MQA task, different models [4, 5, 6, 9] have been proposed to introduce graph-based networks into the traditional Sequence-to-sequence (Seq2Seq) framework to encode the multi-hop information. However, some of the most recent work has shown that the graph structure may not be necessary, and can be replaced with Transformers or proper use of large pre-trained models for multi-hop QA [10, 11]. This motivates us to explore Transformer-based architectures for the relational reasoning requirements of the multi-hop QG (MQG) task.

Another limitation of previous works is that they aim to model \( P(\text{Question}|\langle \text{Answer}, \text{Context} \rangle) \), and ignore the strong constraint of \( P(\text{Answer}|\langle \text{Question}, \text{Context} \rangle) \). As suggested by [2], QA and QG are dual tasks that can help each other. We argue that introduction of a multi-hop QA module can also help MQG.

In this paper, we propose QA4QG, a QA-augmented BART-based framework for MQG. We augment the standard BART framework with an additional multi-hop QA module, which takes the reverse input of the QG system (i.e., question \( Q \) and context \( C \) as input), to model the multi-hop relationships between the question \( Q \) and the answer \( A \) in the given context \( C \). QA4QG outperforms all state-of-the-art models on the multi-hop dataset HotpotQA, with an increase of 8 BLEU-4 and 8 ROUGE points compared to the best results reported in previously published work. Our work suggests the necessity to introduce pre-trained language models and QA modules for the MQG task.

2. RELATED WORK

Most previous approaches on MQG have tried to extend the existing Seq2Seq framework for single-hop QG with reasoning ability. One branch of work models text as graph structure [5, 6] and incorporates graph neural networks into the traditional Seq2Seq framework, mostly the encoder. While this graph-based approach is very intuitive, it relies on additional modules such as semantic graph construction, name entity recognition (NER) and entity linking (NEL), which make

\[ 1 \text{The context can be either sentences or paragraphs} \]
the whole framework complicated and fragile.

Another branch of work on MQG [6,7,12] focuses more on the decoder and aims to augment the Seq2Seq framework with extra constraints to guide the generation. [8] employ multi-task learning with the auxiliary task of answer-related supporting sentences prediction. [7] integrate reinforcement learning (RL) with answer-related syntactic and semantic metrics as reward. The closest effort to our QA4QG is by [12], who introduce a QA-based reward based on SpanBERT in their RL-enhanced Seq2Seq framework, to consider the answerability of the generated question.

On the other hand, the most recent work [9] [10] has shown the strong capability of simple architecture design with large pre-trained language models for multi-hop QA. Such approaches have outperformed the graph network based methods and achieved comparable performance with state-of-the-art architectures such as HGN [13]. This inspires us to explore large pre-trained models for MQG.

3. METHODOLOGY

Our framework, QA4QG, consists of two parts, a BART module and a QA module, as shown in Fig 1. The QA module takes context C and question Q as input, and outputs the probability of each token being the answer. The BART module takes the concatenation of the context C and the answer A, together with the output probability from the QA module as input and generates the question Q token-by-token.

3.1. BART

We choose BART as our backbone for Seq2Seq model because of its outstanding performance on many generation tasks [17]. BART is a Transformer-based model that consists of an encoder and a decoder. The encoder encodes the concatenation of the context C and the answer A. We denote the encoded final representation of the encoder as \( h_{enc} \). Partial structure of the BART decoder is detailed as follow:

\[
H_i^e = \text{MultiHeadAttention}(H_i, H_i, H_i) \quad (1)
\]

\[
H_i^h = \text{Norm}(H_i + H_i^e) \quad (2)
\]

\[
H_i^c = \text{MultiHeadAttention}(H_i^h, h_{enc}, h_{enc}) \quad (3)
\]

where \( H_i \) is the representation for the \( i \)-th layer.

3.2. Answer Relevance Attention

To model the strong relationships of \( P(A|C, Q) \), we propose answer relevance attention, to indicate the answer relevance of each token in context to the target question. Our answer relevance attention can be either soft or hard.

3.2.1. Soft attention

Soft attention can be employed when the ground truth question is available (e.g., in the training phase), and we propose to use a QA module to derive the answer relevance attention. The QA module takes the concatenation of the context \( C \) and question \( Q \) as input, and outputs the prediction of the start and end spans of the potential answer in the context. Specifically, it outputs two probability distributions over the tokens in the context: \( P_{\text{ans}}^s \) and \( P_{\text{ans}}^e \), where \( P_{\text{ans}}^s / P_{\text{ans}}^e \) is the probability that the \( i \)-th token is the start/end of the answer span in context \( C \). The answer relevance attention score \( A_{\text{soft}} \) is calculated via

\[
A_{\text{soft}} = P_{\text{ans}}^s + P_{\text{ans}}^e \quad (4)
\]

where \( A_{\text{soft}} = \{ a_i \}, a_i \) denotes the answer relevance of the \( i \)-th token in context to the question.

For the QA module of our MQG task, we choose the Hierarchical Graph Network (HGN) [13] as it achieves the state-of-the-art performance on the HotpotQA dataset. We believe the \( A_{\text{soft}} \) generated by the HGN model\(^2\) when trained to answer multi-hop question, can naturally learn the answer-aware multi-hop information related to the question inside the context \( C \). This information can then complement the BART model for MQG. Note that other QA models can also be adopted in our framework.

3.2.2. Hard attention

Hard attention can be employed when no question is available (e.g., in the testing phase). Hard attention is inspired by the answer tagging technique from previous work on single-hop QG. Specifically, we first match the answer span with the context \( C \). We then assign a pre-defined score \( p_y \) to the matched tokens, and \( p_n \) to the remaining tokens, to indicate the binary relevance of each token in the context to the answer (in our work, \( p_y = 1.0, p_n = 0.0 \)). We denote hard attention as \( A_{\text{hard}} \).

\(^2\)We use the RoBERTa-large based HGN model, trained on the HotpotQA dataset, and released by the author via https://github.com/youwfan/HGN
Table 1. Comparison between QA4QG and previous MQG methods on the HotpotQA dataset in different encoder input settings. QA4QG outperforms the best models up to 8 BLEU-4 and 8 ROUGE points.

4. EXPERIMENTAL SETUP

To evaluate the effectiveness of our QA4QG framework, we conduct experiments on the HotpotQA [3] dataset, a challenging dataset which contains ~10k multi-hop questions derived from two Wikipedia paragraphs, and requiring multi-hop reasoning to answer. For fair comparison, we follow the data splits of [4,7] to get 90,440 training examples and 6,072 test examples respectively. However, we use the original training data as in HotpotQA, in which each question is paired with two long documents, without pre-processing. [4,7] pre-process the data and only use golden supporting sentences.

4.1. Training Settings

We adopt the BART implementations from Huggingface\(^3\), and experiment based on both the BART-base and BART-large Seq2Seq fine-tuning settings. We run the experiments on single V100 with 16G memory. The maximum source sequence length is set to 512 and 200 respectively, for the full document input and supporting sentences input settings. The training batch size is 6 and 16 respectively for the QA4QG-base and QA4QG-large model, with gradient accumulation steps of 4. We train all model with maximum 5 epochs. The learning rate is 3e-5. During inference, we use beam search with beam size of 4, and we set the maximum target length to 32 and use the default value of the minimum target length, which is 12, with a length penalty of 1.0.

4.2. Baselines

We include the previous work for MQG, and two strong conventional QG models as baselines for comparison: ASs2s-a [14] proposes to decode questions from an answer-separated passage encoder with a new module termed keyword-net, to utilize the information from both the passage and the target answer. SemQG [15] proposes two semantics-enhanced rewards from question paraphrasing and QA tasks to regularize the QG model for generating semantically valid questions. F + R + A [12] uses reinforcement learning during training, and with two strong conventional QG models as baselines for comparison: ASs2s-a [14] proposes to decode questions from an answer-separated passage encoder with a new module termed keyword-net, to utilize the information from both the passage and the target answer. SemQG [15] proposes two semantics-enhanced rewards from question paraphrasing and QA tasks to regularize the QG model for generating semantically valid questions. F + R + A [12] uses reinforcement learning during training, and

\(^3\)https://github.com/huggingface/transformers
First for Women is a woman’s magazine published by Bauer Media Group in the USA. The magazine was started in 1989. It was designed to appeal to women aged 18–34, who were intended as the target audience. The magazine was originally pitched to advertisers, and Pratt was the founding editor of each issue. It was intended to be a glossy magazine based in Englewood Cliffs, New Jersey, but she was voted down by everyone else involved in the making of the magazine. Jane Pratt was the American magazine created to appeal to women who did not like the typical women’s magazine format. Its target audience (pitched to advertisers) was the women who chose to engage with women’s magazines.

Table 2. Ablation study on the QA module. The bottom section uses the supporting sentences (sp) as input.

| Models          | BLEU-4 | METEOR | ROUGE-L |
|-----------------|--------|--------|---------|
| QA4QG-large     | 21.21  | 25.53  | 42.44   |
| w/o QA          | 19.32  | 24.65  | 40.74   |
| QA4QG-base      | 19.68  | 24.35  | 40.44   |
| w/o QA          | 17.43  | 23.16  | 38.23   |
| QA4QG-large (sp)| 25.70  | 27.44  | 46.47   |
| w/o QA          | 25.69  | 27.20  | 46.30   |

Table 2 shows the comparison results between our methods and several state-of-the-art MQG models.

The top section represents using the supporting sentences as input, which is a simplified version of the task. Supporting facts annotations require expensive human labeling and are not always available in the realistic MQG scenario. However, this is the setting used in previous works since their methods can not deal with long documents. QA4QG-large w/o QA shows the performance as in the full documents setting. This actually matches with our intuition. Since there are only two short sentences as context in the supporting documents setting, it is much easier for the QA module to generate the question, the extra improvement from an QA module may not that large.

We then study the effect of the hyper-parameter $\alpha$ in Eq. 6 with different combinations of soft and hard attention during training. The curves of the three metrics in Fig. 2 show that, in general, the more $A_{soft}$, the greater performance improvement QA4QG can achieve. This matches our intuition, since $A_{soft}$ incorporates the question-related multi-hop information into the context via the QA module, while $A_{hard}$ only encodes the explicit answer information. The mixture of both when $\alpha = 0.3$ also yields good results, possibly because of the disparity between training and testing, since during testing we only have $A_{hard}$.

We visualize the attention weights of $A_{soft}$ of an example from the dataset in Fig. 3. As we see, the $A_{soft}$ emphasis on the sentence that contains the multi-hop information ‘First for Women’ and ‘Jane’ in the context, which then constrains the generation model.

5. RESULTS AND ANALYSIS

Table 1 shows the comparison results between our methods and several state-of-the-art MQG models.

The top section represents using the supporting sentences as input, which is a simplified version of the task. Supporting facts annotations require expensive human labeling and are not always available in the realistic MQG scenario. However, this is the setting used in previous works since their methods can not deal with long documents. QA4QG-large w/o QA shows the performance as in the full documents setting. This actually matches with our intuition. Since there are only two short sentences as context in the supporting documents setting, it is much easier for the QA module to generate the question, the extra improvement from an QA module may not that large.

We then study the effect of the hyper-parameter $\alpha$ in Eq. 6 with different combinations of soft and hard attention during training. The curves of the three metrics in Fig. 2 show that, in general, the more $A_{soft}$, the greater performance improvement QA4QG can achieve. This matches our intuition, since $A_{soft}$ incorporates the question-related multi-hop information into the context via the QA module, while $A_{hard}$ only encodes the explicit answer information. The mixture of both when $\alpha = 0.3$ also yields good results, possibly because of the disparity between training and testing, since during testing we only have $A_{hard}$.

We visualize the attention weights of $A_{soft}$ of an example from the dataset in Fig. 3. As we see, the $A_{soft}$ emphasis on the sentence that contains the multi-hop information ‘First for Women’ and ‘Jane’ in the context, which then constrains the generation model.

6. CONCLUSION

In this paper, we propose a novel framework, QA4QG, a QA-augmented BART-based framework for MQG. It is the first work to explore large pre-trained language models for MQG and takes advantage of an additional Multi-hop QA module to further constrain the question generation. Our results on the HotpotQA dataset show that QA4QG outperforms all state-of-the-art models, with an increase of 8 BLEU-4 and 8 ROUGE points compared to the best results previously reported. Our work suggests the advantage of introducing pre-trained language models and QA modules for the MQG task.
7. REFERENCES

[1] Kaichun Yao, Libo Zhang, Tiejian Luo, Lili Tao, and Yanjun Wu, “Teaching machines to ask questions,” in Proceedings of the 27th International Joint Conference on Artificial Intelligence. AAAI Press, 2018, pp. 4546–4552.

[2] Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou, “Question answering and question generation as dual tasks,” arXiv preprint arXiv:1706.02027, 2017.

[3] Zhilin Yang, Peng Qi, Saizheng Zhang, Youshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning, “Hotpotqa: A dataset for diverse, explainable multi-hop question answering,” in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 2369–2380.

[4] Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan, “Semantic graphs for generating deep questions,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, July 2020, pp. 1463–1475, Association for Computational Linguistics.

[5] Dan Su, Yan Xu, Wenliang Dai, Ziwei Ji, Tiezheng Yu, and Pascale Fung, “Multi-hop question generation with graph convolutional network,” in Findings of the Association for Computational Linguistics: EMNLP 2020, Online, Nov. 2020, pp. 4636–4647, Association for Computational Linguistics.

[6] Jianxing Yu, Xiaojun Quan, Qinliang Su, and Jian Yin, “Generating multi-hop reasoning questions to improve machine reading comprehension,” in Proceedings of The Web Conference 2020, New York, NY, USA, 2020, WWW ’20, p. 281–291, Association for Computing Machinery.

[7] Liuyin Wang, Zihan Xu, Zibo Lin, Haitao Zheng, and Ying Shen, “Answer-driven deep question generation based on reinforcement learning,” in Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain (Online), Dec. 2020, pp. 5159–5170, International Committee on Computational Linguistics.

[8] Deepak Gupta, Hardik Chauhan, Ravi Tej Akella, Asif Ekbai, and Pushpak Bhattacharyya, “Reinforced multi-task approach for multi-hop question generation,” in Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 2760–2775.

[9] Devendra Singh Sachan, Lingfei Wu, Minmaya Sachan, and William Hamilton, “Stronger transformers for neural multi-hop question generation,” arXiv preprint arXiv:2010.11374, 2020.

[10] Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu, “Is graph structure necessary for multi-hop question answering?,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 7187–7192.

[11] Dirk Groeneveld, Tushar Khot, Ashish Sabharwal, et al., “A simple yet strong pipeline for hotpotqa,” arXiv preprint arXiv:2004.06753, 2020.

[12] Yuxi Xie, Liangming Pan, Dongzhe Wang, Min-Yen Kan, and Yansong Feng, “Exploring question-specific rewards for generating deep questions,” in Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 2534–2546.

[13] Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu, “Hierarchical graph network for multi-hop question answering,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, Nov. 2020, pp. 8823–8838, Association for Computational Linguistics.

[14] Yanghoon Kim, Hwanhee Lee, Jongbo Shin, and Kyomin Jung, “Improving neural question generation using answer separation,” in Proceedings of the AAAI Conference on Artificial Intelligence, 2019, vol. 33, pp. 6602–6609.

[15] Shiyue Zhang and Mohit Bansal, “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), 2019, pp. 2495–2509.

[16] Jianxing Yu, Wei Liu, Shuang Qiu, Qinliang Su, Kai Wang, Xiaojun Quan, and Jian Yin, “Low-resource generation of multi-hop reasoning questions,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6729–6739.

[17] Mike Lewis and Yinhan et. al Liu, “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” in Proceedings of the 58th ACL.