Generating Highly Relevant Questions

Jiazuo Qiu and **Deyi Xiong***
School of Computer Science and Technology, Soochow University, Suzhou, China
qjzhzw@163.com; dyxiong@suda.edu.cn

Abstract

The neural seq2seq based question generation (QG) is prone to generating generic and undiversified questions that are poorly relevant to the given passage and target answer. In this paper, we propose two methods to address the issue. (1) By a partial copy mechanism, we prioritize words that are morphologically close to words in the input passage when generating questions; (2) By a QA-based reranker, from the n-best list of question candidates, we select questions that are preferred by both the QA and QG model. Experiments and analyses demonstrate that the proposed two methods substantially improve the relevance of generated questions to passages and answers.

1 Introduction

Question generation is to generate a valid and fluent question according to a given passage and the target answer. In general, the answer is a span of words in the passage. QG can be used in many scenarios, such as automatical tutoring systems, improving the performance of QA models and enabling chatbots to lead a conversation.

In the early days, QG has been tackled mainly via rule-based approaches (Mitkov and Ha, 2003; Heilman and Smith, 2010). These methods rely on many handcrafted rules and templates. Constructing such rules is time-consuming and it is difficult to adapt them to other domains.

Very recently, neural networks have been used for QG. Specifically, the encoder-decoder seq2seq model (Du et al., 2017; Zhou et al., 2017; Song et al., 2018) is used to encode a passage and generate a question corresponding to the answer.

Such end-to-end neural models are able to generate better questions than traditional rule-based approaches. However, one issue with the current neural models is that the generated questions are not quite relevant to the corresponding passages and target answers. In other words, the neural QG models tend to generate generic questions (e.g., "what is the name of...?").

In this paper, we propose two methods to deal with this low-relevance issue for QG. First, we present a partial copy method to enhance the existing copy mechanism so that the QG model can not only copy a word as an entire unit from the passage to the generated question but also copy a part of a word (e.g., "start" in "started") to generate a new morphological form of the word in the question. The fine-grained partial copy mechanism enables the QG model to copy morphologically changed words from the passage, increasing the relevance of the generated question to the passage by sharing more words in different forms. Second, we propose a QA-based reranker to rerank QG results. Particularly, we use a neural QA model to evaluate the quality of generated questions, and rerank them according to the QA model scores. Normally, generic questions get low scores from the neural QA model, hence the reranker is able to select non-generic and highly relevant questions.

While alleviating the low-relevance issue, the proposed two methods alone and their combination have also improved the quality of generated questions in terms of both BLEU and METEOR in our experiments.

2 Methods

2.1 The Partial Copy Mechanism

The conventional copy mechanism (Gu et al., 2016; See et al., 2017) can allow the decoder to copy a word from the input passage to the generated question. However, such copy mechanism works at the word level. It cannot copy a part of a word to reproduce an appropriate form of the

*Corresponding author

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 5983–5987, Hong Kong, China, November 3–7, 2019.

word in the generated question. In other words, it cannot copy words with morphological changes. However, morphological changes frequently happen when we transform a passage into a question as grammatical functions of some words (e.g., verbs, nouns, adjectives, etc) change.

Let's consider the following example:

> Passage: Teaching started in 1794.
> Question: When did teaching start?

The morphological variants "start", "starts" of "started" in the passage may be used in the generated question.

In order to encourage the decoder to copy the inflected forms of a word from the passage to the generated question, we propose a partial copy mechanism that measures the character overlap rate of an original word in the passage with its copied form in the question. We first detect the overlapped subsequence of characters between words w_1 and w_2 according to the longest common subsequence (LCS) between them. For example, "start" and "started" have LCS "start". Then we calculate the overlap rate C between w_1 and w_2 as follows.

$$C = \frac{|LCS| * 2}{|w_1| + |w_2|} \quad (1)$$

According to this formula, the overlap rate between "start" and "started" is 0.71. The value range of C is $\in [0, 1]$. Full-word copy is of course encouraged as the value of C is 1.

One problem with C is that many unrelated words also have LCS. For example, "a" is the LCS of "append" and "start". The overlap rate of the two words is 0.18 instead of 0. To ensure the effectiveness of the method, we compute the final overlap rate with a threshold γ :

$$C = \begin{cases} C, & \text{if } C >= \gamma \\ 0, & \text{if } C < \gamma \end{cases} \quad (2)$$

Whenever we generate a word in the question, we find its corresponding word in the passage with the highest attention weight. We then calculate the overlap rate C between the two words and use C to re-adjust the probability of the generated word as follows:

$$P_{adj} = P * (1 + \lambda_1 * C) \quad (3)$$

where P is the original generation probability output by the decoder, λ_1 is a hyperparameter whose range is [0, $+\infty$). We normalize all these re-adjusted probabilities to get the final probability distribution.

2.2 The QA-Based Reranking

Since we use the beam search algorithm in the neural QG decoder, we can generate multiple question candidates. We find that the generated question with the highest probability according to the baseline neural QG model is not always the best question.

Partially inspired by Li et al. (2016), we propose a QA-based reranker to rerank the n-best questions generated by the baseline decoder.

In general, the task of the QG model is to estimate the probability $P(q|p, a)$ of a generated question q given the passage p and target answer a. In the QA-based reranker, we re-estimate the quality of a candidate question by calculating the probability of the target answer a given the passage and the generated question q, i.e., $P(a|p, q)$.

In theory, we can combine the two probabilities to rerank generated questions. But in practice, we take a more intuitive and straightforward way to use the F_1 score of a predicted answer by the trained QA model according to the generated question. The F_1 score is calculated at the character level by comparing the generated answer and gold answer (considering the answers as a set of characters). The idea behind this is that a good question allows the QA model to easily find an answer close to the ground truth answer.

The score used to rerank a question candidate is therefore computed as:

$$score = (1 - \lambda_2) * score_1 + \lambda_2 * score_2 \quad (4)$$

where $score_1$ is the log probability of the candidate question estimated by the baseline QG model, $score_2$ is the F_1 score of the predicted answer by the QA model, and λ_2 is a hyperparameter whose range is [0, 1].

3 Experiments

3.1 Datasets

Following previous work, we conducted our experiments on SQuAD (Rajpurkar et al., 2016), a QA dataset which can also be used for QG. The dataset contains 536 articles and over 100k questions. Since the test set is unavailable, Du et al. (2017) randomly divide the raw dataset into

models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Du et al. (2017)	43.09	25.96	17.50	12.28	16.62
Song et al. (2018) (reported in paper)	–	–	–	13.98	18.77
Song et al. (2018) (our re-running)	42.15	27.21	19.53	14.56	19.15
partial copy (λ_1=0.5)	44.13	28.29	20.17	14.95	19.81
partial copy (λ_1=1)	44.33	28.34	20.17	14.95	20.00
partial copy (λ_1=2)	42.34	26.95	19.15	14.16	**20.16**
QA-based reranking (λ_2=0.2)	42.64	27.57	19.77	14.72	19.43
QA-based reranking (λ_2=0.5)	42.50	27.42	19.62	14.56	19.38
QA-based reranking (λ_2=0.8)	42.43	27.34	19.54	14.48	19.34
partial copy + QA-based reranking	**44.61**	**28.78**	**20.59**	**15.29**	20.13

Table 1: Experiment results on the test set.

train/dev/test set. In our experiments, we used the same data split as Du et al. (2017).

3.2 Baseline and Settings

Our baseline is based on the QG model proposed by Song et al. (2018). To be specific, it is a seq2seq model with attention and copy mechanism. The model consists of two encoders and a decoder. The two encoders encode a passage = (p_1,...,p_M) and an answer = (a_1,...,a_N) respectively. Additionally, multi-perspective matching strategies are used to combine the two encoders. With information from the encoders, the decoder generates a question word by word.

We retained the same values for most hyperparameters in our experiments as the baseline system (Song et al., 2018). We used Glove (Pennington et al., 2014) to initialize the word embeddings and trained the model for 10 epochs. Copy and coverage mechanism (See et al., 2017) were included while additional lexical features (POS, NER) were not. We used adam (Kingma and Ba, 2015) as the optimizer during training. The beam size was set to 20 for the decoder.

In the experiments of the partial copy mechanism, we set the threshold γ to 0.7. Three values (0.5, 1 and 2) were tried for λ_1.

In the experiments of the QA-based reranking, we used the SAN model (Liu et al., 2018) as the QA model. 0.2, 0.5 and 0.8 were tried for λ_2.

We used automatic evaluation metrics: BLEU-1, BLEU-2, BLEU-3, BLEU-4 (Papineni et al., 2002) and METEOR (Denkowski and Lavie, 2014). The calculation script was provided by Du et al. (2017).

3.3 Results

Experiment results are shown in Table 1, from which we observe that our methods substantially improves the baseline in terms of all evaluation metrics.

The combination of the proposed two methods (λ_1 = 1, λ_2 = 0.2) achieve the best performance, gaining 0.73 BLEU-4 and nearly 1 METEOR point of improvements over the baseline (obtained by re-running the source code of the baseline, higher than the results reported in the paper (Song et al., 2018)). The proposed partial copy mechanism alone obtain substantial improvements over the baseline, especially in terms of BLEU-1 and BLEU-2. This is because this method is able to help the decoder copy morphologically changed words from passages. The application of the QA-based reranking obtain further improvements over the partial copy mechanism, indicating that the two methods are complementary to each other.

4 Analysis

Partial copy: We use the method to enhance the existing copy mechanism, making generated questions more relevant to passages and target answers. The average proportion of words (or their other morphological forms) that are copied from the passages in generated questions increases from 75.49% (the baseline) to 78.74%. Under such a mechanism, generic questions such as "what is the name of...?" will be penalized by our method as the overlap rate between words in these generic questions and those in passages is low. On the contrary, questions with higher overlap rate and therefore higher relevance to the input passage are rewarded by the new copy mechanism. In order to testify this hypothesis, we counted the numbers

Question templates	Song et al. (2018)	Our model
What is/was the name of ...?	8,180/13,740	5,618/9,990
What type of ...?	4,942	3,805
What is/was another name ...?	2,731/26	57/3
What is/was the total ...?	470/794	101/62
What is/was it ...?	251/57	111/32

Table 2: Frequency of generic questions generated by the baseline and our methods.

Passage: in the polytechnic sector : wellington polytechnic amalgamated with massey university .
Answer: wellington polytechnic
Question: what school did massey university combine with ?
Baseline: what is the name of the polytechnic sector in the polytechnic ?
Partial copy: who amalgamated with massey university in the polytechnic sector ?

Passage: in practice , catholic services in all provinces were quickly forbidden , and the reformed church became the " public " or " privileged " church in the republic .
Answer: catholic services
Question: what was forbidden in all provinces ?
Baseline: what was the " public " church in the republic ?
QA-based reranking: what was forbidden in all provinces in the republic ?

Table 3: QG examples.

of such generic questions generated by the baseline and our method, which are shown in Table 2. It is clearly seen that the number of generic questions is significantly decreased after the partial copy mechanism is used.

In the first example displayed in Table 3, the phrase "what school" is difficult to be generated as it does not appear in the input passage. The baseline model generates the generic question "what is the name of...?", which is not relevant to the passage and target answer. Such generic questions are generated because the templates of these questions occur frequently in the training data. The trained seq2seq model is prone to generating these "safe" questions, similar to the undiversified response generation in seq2seq-based dialogue model (Li et al., 2016). In contrast, our model is able to generate a more relevant question including a rare word "amalgamated" as the word has a high overlap rate.

QA-based reranking: In our experiments, a total of 2,099 questions were reranked. Among them, 1,117 examples achieve a higher BLEU-4 score after reranking, while only 821 examples have a lower BLEU-4 after reranking.

In the second example of Table 3, the question with the highest score generated by the baseline is "what church", while the ground truth question is asking "what was forbidden". Since the generated questions to be reranked are different to each other, the QA model naturally finds different answers to these questions. For "what church", the QA model detected "reformed church" as the answer while for "what was forbidden", the QA model correctly detected the target answer "catholic services". Therefore, the QA-based reranker is able to find the answer-relevant questions.

5 Related Work

The neural QG is an emerging task. Unlike the extractive QA, most neural QG models are generative. Du et al. (2017) pioneer the neural QG by proposing neural seq2seq models to deal with the task. Unfortunately, they do not use the target answer for QG. At the same time, Zhou et al. (2017) present a similar model for QG. They use answer position embeddings to represent target answers and explore a variety of lexical features.

After that, many QG studies have been conducted on the basis of the widely-used seq2seq architecture together with the attention and copy mechanism. Song et al. (2018) propose two encoders for both the passage and the target answer. Du and Cardie (2018) employ coreferences as an additional feature. Kim et al. (2019) propose a model of answer separation. Yuan et al. (2017) and Kumar et al. (2018) adopt reinforcement learning to optimize the generation process.

QA and QG are closely related to each other. Tang et al. (2017) treat QA and QG as dual tasks, and many other studies use QG to enhance QA or jointly learn QG and QA (Duan et al., 2017; Wang et al., 2017; Sachan and Xing, 2018).

6 Conclusion

In this paper, we have presented two methods to improve the relevance of generated questions to the given passage and target answer. Experiments and analyses on SQuAD show that both the partial copy mechanism and QA-based reranking improve the relevance of generated questions in terms of both BLEU and METEOR.

Acknowledgements

The present research was supported by the National Natural Science Foundation of China (Grant No. 61622209). We would like to thank the anonymous reviewers for their insightful comments.

References

Michael J Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. pages 376–380.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. *meeting of the association for computational linguistics*, 1:1907–1917.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *meeting of the association for computational linguistics*, 1:1342–1352.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. pages 866–874.

Jiatao Gu, Zhengdong Lu, Li Hang, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. pages 609–617.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. *national conference on artificial intelligence*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *international conference on learning representations*.

Vishwajeet Kumar, Ganesh Ramakrishnan, and Yuanfang Li. 2018. A framework for automatic question generation from text using deep reinforcement learning. *arXiv: Computation and Language*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. *north american chapter of the association for computational linguistics*, pages 110–119.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018. Stochastic answer networks for machine reading comprehension. *meeting of the association for computational linguistics*, 1:1694–1704.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. pages 17–22.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *Proc Meeting of the Association for Computational Linguistics*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. pages 1532–1543.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *empirical methods in natural language processing*, pages 2383–2392.

Mrinmaya Sachan and Eric P Xing. 2018. Self-training for jointly learning to ask and answer questions. 1:629–640.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *meeting of the association for computational linguistics*, 1:1073–1083.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018. Leveraging context information for natural question generation. 2:569–574.

Duyu Tang, Nan Duan, Tao Qin, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv: Computation and Language*.

Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. A joint model for question answering and question generation. *arXiv: Computation and Language*.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. 2017. Machine comprehension by text-to-text neural question generation. *meeting of the association for computational linguistics*, pages 15–25.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. *arXiv: Computation and Language*, pages 662–671.