Question Generation using Knowledge Graphs with the T5 Language Model and Masked Self-Attention

Kosuke Aigo, Takashi Tsunakawa, Masafumi Nishida, Masafumi Nishimura
Graduate School of Integrated Science and Technology
Shizuoka University
Shizuoka, Japan
Email: aigo.kosuke.16@shizuoka.ac.jp
Email: {tuna, nishida, nisimura}@inf.shizuoka.ac.jp

Abstract—**Question generation is helpful for understanding reading comprehension, spontaneous questioning in chatting systems, and expanding datasets for answering questions. In previous studies, many models have been used to generate questions from contexts, but none was suitable in large-length contexts. To overcome this challenge, we generated questions from an intermediate representation of a context, such as knowledge graphs. In this study, we focused on developing questions using knowledge graphs with the T5 language model. We used the language model to create questions using the knowledge graph and mask the self-attention of the encoder to train the model by explicitly preserving the graph's structure. As a result of the automatic evaluation, the T5 language model with and without mask was comparable with the bidirectional Graph2Seq model (G2S), known as the QG model, using knowledge graphs. Moreover, the masked language model was slightly better than the non-masked model in t5-small on four benchmarks. The code and data are publicly available at https://github.com/Macho000/T5-for-KGQG.**

Keywords—**KG2QG, Question generation, T5 language model.**

I. INTRODUCTION

When we study or read something, we often try to assess our comprehension. In schools, the qualitative assessment of our learning is performed via tests. However, when we undertake self-studies or read certain materials, self-assessment through comprehension tests is difficult. Therefore, an automatic question generation for assessing our level of understanding is needed. Moreover, if the system can spontaneously ask questions in a chatting dialog system, it could lead to a good and compelling conversation. Further, question generation is useful in augmenting a data for question answering.

Research on question generation includes generating questions from contexts and intermediate representations such as knowledge graphs. Studies on generating questions from contexts include using the language model BERT [1] and a unique model without a language model [2]. Unfortunately, these models are inefficient in generating questions when the length of an input sequence is large. Therefore, converting the input sequence into intermediate representations such as knowledge graphs is necessary, where the knowledge graphs can then be used to generate questions.

In [3] and [4], knowledge graphs have been used to generate questions. In [3], the authors use a multihop question generation system called MHQG+AE and a transformer-based model to generate questions using knowledge graphs. The MHQG+AE considers a knowledge graph as a set of tuples (Subject, Predicate, Object) and does not explicitly use the graph structure. To overcome this, Chen et al. [4] proposed a bidirectional Graph2Seq model ($G2S$) to create questions by explicitly preserving the graph structure using a graph neural network (GNN). Their results were better than that of MHQG+AE.

The $G2S$ model encodes the knowledge graph in both forward and backward directions using GNNs. Moreover, this model directly copies nodes from the knowledge graph using a node-level copying mechanism and provides generated answers. Although the graph structure is preserved, information on the essential nodes is not considered in the $G2S$ model. Notably, none of the previous studies employed the language model for KG-QG tasks.

In recent years, transfer learning, which fine-tunes pretrained language models such as BERT and T5, has shown great success in named-entity recognition, question answering, and sentence classification. For example, the T5 model [5] has succeeded in sentence summarization and question answering by performing a word prediction task in pretraining and predicting the sequences of two or more words in phrases.

Therefore, we anticipate that using a knowledge graph-based language model to the generate questions will also produce great results in the KG-QG task. This is because [6] has shown that proper masking attention can be effective in question answering, natural language inference, sentiment analysis, and document ranking. Therefore, we applied the language model with and without the masking of the encoder's self-attention to the KG-QG task for preserving the graph structure.

II. APPROACH

Our approach is to use the T5 language model [5] (w/o mask model) and masking self-attention model in the encoder (w/ mask model) to generate questions based on knowledge graphs. We used T5-small (t5small) and T5-base (t5base) to determine the effects of the language model size on the results. We used 60 and 220 million parameters with their corresponding attention heads of 6 and 12, respectively. Furthermore, we

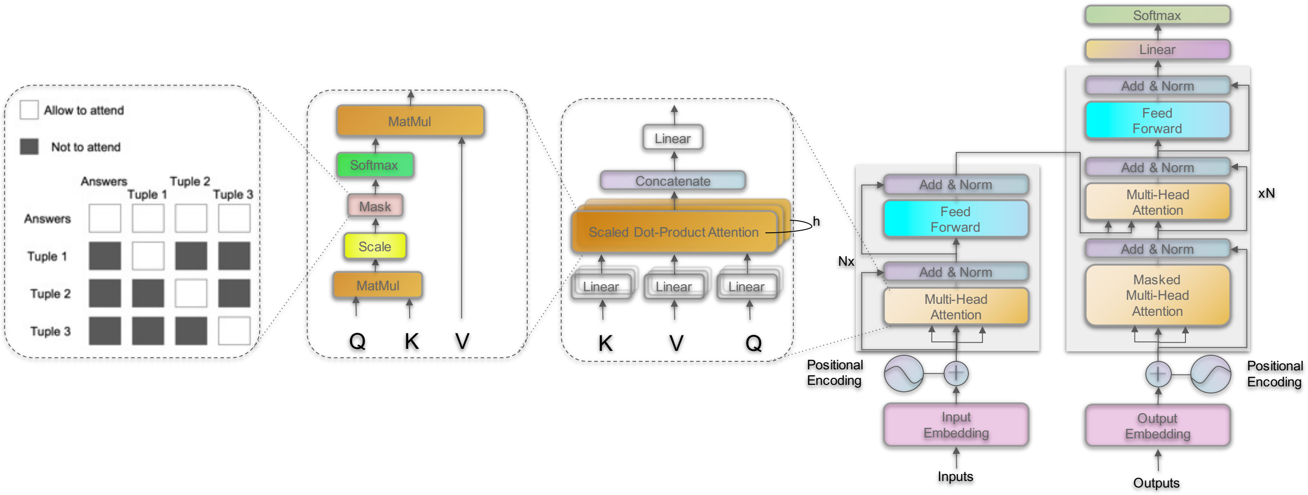

Fig. 1. Overall architecture w/ mask model.

TABLE I
Automatic evaluation results on WQ and PQ.

Method	WQ			PQ		
	BLEU	METEOR	ROUGE	BLEU	METEOR	ROUGE
G2S (baseline) [4]	**29.45**	**30.96**	**55.45**	**61.48**	**44.57**	**77.72**
t5small w/ mask	25.42	28.53	52.49	55.85	42.09	74.52
t5small w/o mask	23.36	28.00	51.35	54.49	42.73	74.71
t5base w/ mask	27.10	29.56	53.59	57.50	43.30	75.63
t5base w/o mask	27.85	30.32	54.65	57.73	43.66	76.01

Algorithm 1 How to mask self-attention in Encoder
$G = (Sub, Pred, Obj)$

```
$Input \leftarrow Answer_1, Answer_2, ..., Answer_k, G_1, G_2, ..., G_n$
$attention\_mask[|Input|][|Input|] \leftarrow 0$
for $i$, $sequence\ in\ enumerate(Input)$ do
    if $sequence == Answer$ then
        $attention\_mask_i \leftarrow 1$            ▷ Focus on all Input
    else
        $attention\_mask_{ii} \leftarrow 1$   ▷ Focus on the same tuple
    end if
end for
```

used the mask self-attention model (w/ mask model) in the encoder to pay attention to the same tuples (Subject, Predicate, Object) to preserve the graph structure as in Algorithm 1.

The overall architecture for w/ mask model is shown in Fig 1. The w/ mask model is masked to pay attention to the same tuple as shown in Fig 1, while the w/o mask model is not masked. In Decoder's masked attention model, the words to be predicted are masked so that they are not visible (causal masking), as employed in the T5 model.

III. Experiments

A. Models

We compared the following three models: w/ mask, w/o mask, and *G2S* (baseline). The w/ mask and w/o mask models were executed using both the t5small and t5base models. In the w/ mask model, the encoder's self-attention was masked to focus on the same tuples (Subject, Predicate, Object) as in Algorithm 1, whereas in the w/o mask model, the encoder's self-attention was not masked. Finally, in the *G2S* model, a bidirectional Graph2Seq model was used.

B. Data and Metrics

Following the approach in [4], we used WebQuestions (WQ) and PathQuestions (PQ) as benchmarks. Note that WQ and PQ are similar, and use Freebase as their underlying knowledge graph. Specifically, each of WQ and PQ is a set of tuples ($\{Q_t$, G, $E_A)\}$), where Q_t is a natural language question; G is the subgraph derived from the question; and E_A is an answer entity toward question Q_t. However, PQ includes a verbalized predicate and entity.

Brief statistics of both datasets are shown in Table II. The WQ dataset uses a combination of WebQuestionSP and Complex WebQuestions. Both WebQuestionSP and Complex WQ contain natural language questions, corresponding SPARQL queries, and answers. [3] created WQ by changing the SPARQL SELECT clause to a CONSTRUCT clause, thereby retrieving the knowledge graph rather than the corresponding SPARQL answer. The PQ is similar to WQ, but PQ contains only linguistic entities and predicates.

Furthermore, we used the BLEU-4, METEOR, and ROUGE-L scores as proposed in [4] to evaluate the results of the model generation. The BLEU-4 and METEOR scores were originally designed for machine translation, and the ROUGE-L score was designed for machine summarization.

TABLE II
DATA STATISTICS. THE MIN/MAX/AVG STATISTICS ARE REPORTED ON THE QUERIES AND KG SUBGRAPH TRIPLES.

Data	# examples	# entities	# predicates	# triples	query length
WQ	22,989	25,703	672	2/99/5.8	5/36/15
PQ	9,731	7,250	378	2/3/2.7	8/25/14

C. Model Settings

The maximum length of the input was set to 512 and the maximum length of the output was set to 100. Therefore, if there were more than 512 tokens in the input, they were excluded. AdamW was used as the optimizer, and the learning rate was set to 3.0×10^{-3}. The number of epochs was set to 8 for all models. For the w/o mask model, the number of batches was set to 8, and for the w/o mask model, the number of batches was set to 2. All experiments were conducted on Google Colab.

IV. RESULTS AND DISCUSSION

Table I shows the comparison of the w/ mask and w/o mask models for the t5small, t5base, and G2S models (baseline).

In comparing the proposed models and the baseline, the performance of t5base w/o mask was similar to that of the baseline. This suggests that the T5 language model, which considers context information, is as effective as GNN in generating questions using knowledge graphs containing graph structures. In other words, the knowledge graph structure is as essential for the KG2QG task as the probability distribution of word sequences. This suggests that masking and new architectures that can better preserve the graph structure are needed for the KG2QG task using the language model.

In comparing the w/ mask and w/o mask models, the w/ mask score was higher in four of the six evaluation metrics for t5-small. This indicates that the w/ mask model in t5-small can learn more efficiently by masking self-attention to focus on the same tuple and understand which node is essential. However, the w/o mask model score was higher in all the six evaluations in the t5-base. This may be that explicitly preserving the graph structure is less effective for models with a large number of parameters, and the model automatically learns the important node.

V. CONCLUSION

In this study, we applied the T5 language model with and without masking to the KG-QG task. The language model results were comparable with the existing models for PQ and WQ. In addition, we compared the language models with and without mask, and our results showed that preserving the graph structure explicitly resulted in better scores in the t5-small model. Future directions include exploring effective ways of using language models to preserve graph structure better.

ACKNOWLEDGMENT

We thank the editor, anony-mous referees, all authors of previous reseach.

REFERENCES

[1] Y. H. Chan and Y. C. Fan, "BERT for question generation," in *Proc. of the 12th International Conference on Natural Language Generation*, 2019, pp. 173–177.

[2] P. Nema, A. K. Mohankumar, M. M. Khapra, B. V. Srinivasan, and B. Ravindran, "Let's ask again: Refine network for automatic question generation," in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019.

[3] V. Kumar, Y. Hua, G. Ramakrishnan, G. Qi, L. Gao, and Y.-F. Li, "Difficulty-controllable multi-hop question generation from knowledge graphs," in *International Semantic Web Conference*. Springer, 2019, pp. 382–398.

[4] Y. Chen, L. Wu, and M. J. Zaki, "Toward subgraph guided knowledge graph question generation with graph neural networks," *arXiv preprint arXiv:2004.06015*, 2020.

[5] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. L. Peter, and J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[6] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.