Knowledge Graph for Efficient Multi-hop Question Generation (Student Abstract)

Al Hasib Mahamud, Yllias Chali

University of Lethbridge
4401 University Dr W
Lethbridge, AB T1K 3M4 Canada
alhasib.mahamud@uleth.ca, yllias.chali@uleth.ca

Abstract

Question generation is the task of natural language processing where the goal is to generate fluent, grammatically correct, error-free questions based on a given input context and optionally an answer. Multi-hop question generation is a more complex task compared to traditional single-hop question generation, as it requires reasoning over multiple information from multiple input contexts in generating multi-hop questions. In this paper, we have addressed the challenge of building a multi-hop question generation system by combining the knowledge graphs with large language models. We have designed a framework KG4QG (Knowledge Graph for Question Generation), where knowledge graphs are generated from the input contexts. For the knowledge graph embedding, we have used Graph Attention Network, and for input text embedding, we have leveraged Sentence Transformer. Finally, we apply BART and T5 models as Large Language Models to generate multi-hop questions from our proposed model. Using HotpotQA dataset to evaluate the performance of our KG4QG framework, our proposed methodology has shown an enhancement of performance over the previous methodologies.

Introduction

In recent years, Question Generation (QG) has been a topic of extensive research, but most of the research is based on the generation of a single question in a single paragraph where the answer of the question depends on a small amount of reasoning (Emerson and Chali 2023). Multi-hop QG can be defined as the task focusing on generating complex questions which require integrating information from multiple interconnected sources, which is called multi-hop reasoning. Multi-hop QG aggregates several evidence from multiple paragraphs, and ensures reasoning over them to generate meaningful, answer-related, factual-coherent questions (Su et al. 2020).

Proposed Model

Sequence-to-sequence based transformer models are already capable of generating simple questions where only one single paragraph is required to answer a logical solution of the question. In this research, our main aim is to see the impact of utilizing Knowledge Graph (KG) in the multi-hop question generation.

Knowledge Graph Creation

To create the knowledge graph, the input text is converted to annotated text by performing coreference resolution on the input text so the redundancy of nodes are removed, and the same entities are linked so per entity one node is created in the graph, and a more accurate graph is generated. First we have to start the Stanford CoreNLP [1] server. The Open Information Extraction (OpenIE) [2] annotator is used as it can extract triplets from the input text. A triplet is a representation of a subject, a relation, and the object of the relation. Finally, the graph is visualized using the Python NetX [3] package.

Graph Representation Creation

After generating the knowledge graph using Stanford CoreNLP, we created a graph representation of each knowledge graph by constructing PyG[4] compatible tensors. The triplets are converted to the Node Features, Edge Index, and Edge Attribute. Node Features are the dictionary mapping where each node (subject or object) of the graph is converted to an index number to convert human readable node labels to machine readable index number. To represent the connection between source to destination of the graph, edge index represents the connectivity. Edge Attribute represents the embeddings for subject to object relations. After generating the attributes from triplets, each row of questions is encoded by the all-Minim-L6-v2 [5] model to generate the space vector of the questions. Finally, the graph object representation is generated by creating a Data object from the PyTorch Geometric (PyG) [6] library. The graph data object is created by packaging the node features, edge index, edge attributes, and the target variable, where the target variable is the question in this research.

[1] https://stanfordnlp.github.io/CoreNLP/

[2] https://stanfordnlp.github.io/CoreNLP/openie.html

[3] https://networkx.org/

[4] https://pyg.org/

[5] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[6] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.data.Data.html

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
MulQG	40.15	26.71	19.73	15.20	35.30	20.51
GNET-QG	49.72	38.95	32.88	27.93	40.25	**49.87**
MultiFactor	**54.17**	41.50	33.74	28.22	28.60	44.17
KG4QG(T5 Backbone)	47.08	39.11	36.32	34.27	41.48	40.41
KG4QG(BART Backbone)	53.38	**44.81**	**41.24**	**38.56**	**47.11**	46.08

Table 1: Performance Comparison between KG4QG and Existing Models

Integrated Embedding Representations

The graph embedding is generated by the graph attention network (GAT) model. We load the graph data object, then the model is trained, and the evaluation is done on a test set from the data object. The training, testing, and validation split is done as 70:15:15 ratio. During training, the Mean Squared Error (MSE) is used to calculate the loss. Early stopping and model checkpoint are also used. The model is trained for 200 epochs.

The input text is converted to a numerical representation using the Sentence Transformer. We utilize the all-Minim-L6-v2 model to generate the space vector of the input text, so the encoded input text is generated. The graph embedding is concatenated at the end of the input text embedding by torch.stack[7].

Encoder

The encoder includes feed-forward neural networks and a multi-head self-attention mechanism. As a backbone, we have utilized BART-base and T5-base both as encoders separately and have tried to find which model works better. To ensure that the text fits the input length of the model and to generate an attention mask for the input text, we have fed tokenization into the LLM encoder component. We have generated the attention mask from the input text and then the combined embedding layer is ready to be fed into the encoder of the LLMs.

Decoder

The LLM decoder which follows the transformer architecture is used for autoregressive generation, that is to say, based on previous predictions, it produces questions token by token. Decoder uses mask self-attention so the model ensures autoregressive masking, so the next prediction of word is based on the words that are previously examined in the sequence plus the current word.

Experiments

Dataset

In this research, we have utilized HotpotQA (Yang et al. 2018) dataset for training and evaluating our model. According to Su et al. (2020), it is important to filter out all yes/no answer based data samples from the HotpotQA dataset focusing on the multi-hop ability so we have removed those samples and also have excluded the supporting sentences from the dataset. After filtering out the dataset, we have partitioned the dataset into three parts: training dataset, validation dataset, and testing dataset according to 70:15:15 ratio.

Fine-Tuning

Fine-tuning is performed to make the pre-trained models task specific of multi-hop question generation. In both BART and T5 models, initially the number of epochs is fixed as 50, where early stopping is used as a regularization technique to prevent the model from overfitting. Here, based on validation loss the patience parameter is fixed as 3, which means if no change occurs in validation loss, after three epochs, the training will be stopped. The best outcome of the T5-base model is achieved after 50 epochs, where for the BART-base model the best outcome is achieved after 20 epochs. The Adam optimizer (Kingma and Ba 2017) is used, where the Cross-Entropy function (Mao, Mohri, and Zhong 2023) is used to calculate the loss. The learning rate is fixed as 10^{-4}.

Evaluation and Conclusion

To evaluate the performance of KG4QG, we have utilized automated evaluation metrics. These metrics are chosen based on the widespread utilization in question generation task. To demonstrate the efficiency of our model KG4QG with BART backbone and T5 backbone, we have done comparison with existing models on multi-hop question generation including MulQG by Su et al. (2020), GNET-QG by Jamshidi and Chali (2025), and MultiFactor by Xia et al. (2023) because of their significance results in evaluation metrics. From Table 1, we can see that our model KG4QG outperforms across all metrics compared to the existing models because of the enhancement of LLMs with Knowledge Graph integration, except on BLEU-1, where the difference with MultiFactor is 0.79, and METEOR, where the difference with GNET-QG is 3.79. Our results show that knowledge graph integration with LLMs is an effective solution for the enhancement of multi-hop QG task.

Further research can be focused on the low-resource multi-hop question generation, enriching cross-lingual and multilingual capacity to improve the ability of multi-hop question generation system, utilizing different prompting techniques like Chain-of-Thought (CoT) prompting (Wei et al. 2023), Tree of Thoughts (ToT) prompting (Yao et al. 2023), Graph of Thoughts (GoT) prompting (Besta et al. 2024) that may improve the performance of LLMs for multi-hop question generation.

[7]https://pytorch.org/docs/stable/generated/torch.stack.html

Acknowledgments

We would like to thank the anonymous reviewers for their useful comments. The research reported in this paper was conducted at the University of Lethbridge and supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the University of Lethbridge.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Janvin, C. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null): 1137–1155.

Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; and Hoefler, T. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16): 17682–17690.

Chaudhary, A.; Rajabi, E.; Kafaie, S.; and Milios, E. 2025. Fact retrieval from knowledge graphs through semantic and contextual attention. *Expert Systems with Applications*, 282: 127612.

Du, H.; Le, Z.; Wang, H.; Chen, Y.; and Yu, J. 2022. COKG-QA: Multi-hop Question Answering over COVID-19 Knowledge Graphs. *Data Intelligence*, 4(3): 471–492.

Emerson, J.; and Chali, Y. 2023. Efficient Multi-hop Question Generation. *Procedia Computer Science*, 222: 217–222.

Jamshidi, S.; and Chali, Y. 2025. GNET-QG: Graph Network for Multi-hop Question Generation. In *Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK)*, 20–26. Abu Dhabi, UAE: International Committee on Computational Linguistics.

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Kor, Y.; Tan, L.; Musilek, P.; and Reformat, M. Z. 2024. Integrating Knowledge Graphs into Distribution Grid Decision Support Systems. *Future Internet*, 16(1).

Mao, A.; Mohri, M.; and Zhong, Y. 2023. Cross-Entropy Loss Functions: Theoretical Analysis and Applications. arXiv:2304.07288.

Mavi, V.; Jangra, A.; and Jatowt, A. 2024. Multi-hop Question Answering. arXiv:2204.09140.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781.

Pietrasik, M.; Reformat, M.; and Wilbik, A. 2024. Hierarchical Blockmodelling for Knowledge Graphs. arXiv:2408.15649.

Qiu, L.; Xiao, Y.; Qu, Y.; Zhou, H.; Li, L.; Zhang, W.; and Yu, Y. 2019. Dynamically Fused Graph Network for Multi-hop Reasoning. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6140–6150. Florence, Italy: Association for Computational Linguistics.

Rao, D. J.; Mane, S. S.; and Paliwal, M. A. 2022. Biomedical Multi-hop Question Answering Using Knowledge Graph Embeddings and Language Models. arXiv:2211.05351.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084.

Su, D.; Xu, Y.; Dai, W.; Ji, Z.; Yu, T.; and Fung, P. 2020. Multi-hop Question Generation with Graph Convolutional Network. In Cohn, T.; He, Y.; and Liu, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4636–4647. Online: Association for Computational Linguistics.

Sun, K.; Wang, J.; Jiang, H.; Hu, Y.; and Yin, B. 2024. Query-Enhanced Adaptive Semantic Path Reasoning for Inductive Knowledge Graph Completion. arXiv:2406.02205.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. arXiv:1409.3215.

Tang, Y.; Liu, T.; Liu, G.; Li, J.; Dai, R.; and Yuan, C. 2019. Enhancement of power equipment management using knowledge graph. In *2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, 905–910. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2023. Attention Is All You Need. arXiv:1706.03762.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Xia, Z.; Gou, Q.; Yu, B.; Yu, H.; Huang, F.; Li, Y.; and Cam-Tu, N. 2023. Improving Question Generation with Multi-level Content Planning. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 800–814. Singapore: Association for Computational Linguistics.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2369–2380. Brussels, Belgium: Association for Computational Linguistics.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T. L.; Cao, Y.; and Narasimhan, K. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. arXiv:2305.10601.

Zamani, H.; Dumais, S.; Craswell, N.; Bennett, P.; and Lueck, G. 2020. Generating Clarifying Questions for Information Retrieval. In *Proceedings of The Web Conference 2020*, WWW '20, 418–428. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370233.

Zhang, J.; Zhang, H.; Zhang, D.; Yong, L.; and Huang, S. 2024. End-to-End Beam Retrieval for Multi-Hop Question Answering. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 1718–1731. Mexico City, Mexico: Association for Computational Linguistics.