KGP-QG: Multi-hop Reasoning with Knowledge Graphs in LLMs

Al Hasib Mahamud[1,*], Yllias Chali[1]

[1]Department of Computer Science, University of Lethbridge, Alberta, Canada

Abstract

Multi-hop question generation is the task of generating complex questions that require reasoning over multiple information from multiple contexts. Though the recent works on multi-hop question generation have shown a great impact with excellent outcomes using sequence-to-sequence models and large language models (LLMs), in this paper, we have shown the impact of knowledge graph-based prompting approach on multi-hop question generation. For this purpose, we have proposed a framework, KGP-QG (Knowledge Graph Based Prompting for Question Generation), where, at first, we have generated knowledge graph from the input contexts. The generated knowledge graph is integrated back into the input context to form a knowledge graph-based prompt. Generated knowledge graph enriched prompt is fed to LLMs, where the prompt is used for fine-tuning in two LLMs- BART and T5 models. For this research, the HotpotQA dataset is used for result evaluation and fine-tuning. The result of our proposed framework, KGP-QG, has outperformed the existing methodology.

Keywords

Knowledge Graph Prompting, Multi-hop Reasoning, Question Generation

1. Introduction

Multi-hop question generation (MHQG) is the task of automatically generating complex questions that require complex reasoning over multiple source inputs. As it is a complex task, in the multi-hop question generation system, a reader needs to understand the contexts, connect the facts by reason over the information from multiple input contexts to generate the answer to the multi-hop question [1]. Single-hop question generation system depends on a single information, where multi-hop question generation system depends on the integration of multiple information from multiple contexts [2].
Large Language Models (LLMs) are effective in performing different natural language processing (NLP) tasks like question generation (QG), question answering (QA), text summarization, etc. But it often struggles to generate accurate output because of the lack of extensive knowledge. This limitation indicates the need for non-parametric knowledge for LLMs [3]. Now 'pre-train, prompt, and predict' paradigm has revolutionized in NLP for the utilization of various real-world applications [4, 5, 6, 7, 8]. To mitigate the gap, we propose a knowledge graph-based prompting approach for multi-hop question generation (KGP-QG), where knowledge graph-based prompting is fed to LLMs with the input contexts. Different prompting approaches are seen in recent works like Tree/Chain/Graph-of-thought to guide LLMs for a specific task [9, 10, 11, 12].The major contribution of this research can be marked as:

- **KG based Prompting Formulation**
  From the HotpotQA[1] dataset, we have generated triplet based KG for each input text and we design knowledge graph based prompting method to guide the LLMs for specific MHQG task.
- **Enhancement of MHQG**
  Our approach, which is based on fine-tuning the LLMs, has shown outperformance compared to the existing methodology for multi-hop question generation task.

Joint Workshop on Statistics and Knowledge Integration for Logic, Learning, Ethical Decisions, and LLMs (SKILLED-LLMs 2026), co-located with FLoC 2026, Lisbon, Portugal
*Corresponding author.
alhasib.mahamud@uleth.ca (A. H. Mahamud); yllias.chali@uleth.ca (Y. Chali)
0009-0001-9813-516X (A. H. Mahamud); 0009-0008-5198-7788 (Y. Chali)

[1]https://huggingface.co/datasets/hotpotqa/hotpot_qa

2. Preliminaries

2.1. Motivation

The main motivation of this research work is utilizing knowledge graph based prompting on the LLMs to generate multi-hop questions. MHQG is already a known topic to researchers as it is a complex task to generate multi-hop questions compared to single-hop questions, which requires multiple reasoning over information. A good amount of research has been done on the multi-hop question generation task based on sequence-to-sequence text generation approach and constructing graphs (Entity Graphs, Knowledge Graphs) to provide more information to LLMs. According to the best of our knowledge, this is the first research work where KG is being utilized for formatting prompting and the prompt is leveraged for fine-tuning the LLMs.

2.2. Knowledge Graph

A knowledge graph can be defined as the collection of triplets [13]. It is a data management system to combine different types of data and graphs are utilized to visualize it [14, 15]. According to Hogan et al. [16], KG can be defined as *" a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities".* The notation of knowledge graph can be represented as:

$$\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}) \tag{1}$$

Where,
$\mathcal{E}$ is the set of entities/nodes.
$\mathcal{R}$ is the set of relations.
$\mathcal{T}$ is the set of triplets which can be represented by Equation 2.

$$\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E} \tag{2}$$

2.3. Hallucination

Hallucination is the factual incorrectness; it can be defined as the generation of fluent, coherent, and reasonable texts where factual or accurate information is lacking, which results in reliability and trustworthiness in LLMs for real-world applications [17, 18]. Mathematically, hallucination occurs when a model provides a higher probability to an ungrounded generated sequence compared to a factually grounded sequence [19].

$$p_\theta(y_{hallucination}|x) > p_\theta(y_{grounded}|x) \tag{3}$$

2.4. Problem Definition

Given a multi-hop input context X, the conditional probability of the output sequence (multi-hop question) $Y = [y_1, y_2, y_3, ..., y_m]$ can be modeled as $P(Y|X) = \prod_{t=1}^{m} P(y_t|y_{<t}, X)$ where each token y_t is generated autoregressively conditioned on the preceding token $y_{<t}$ and the full context X. As we are generating prompt which consists of Instruction, Knowledge Graph, and the input Context, so X can be defined as Equation 4.

$$X = \texttt{Instruction} \oplus \texttt{KG}_{Triplets} \oplus \texttt{Context} \tag{4}$$

Where, $\oplus$ indicates the concatenation with special separator token
$\texttt{KG}_{Triplets} = \sum_{i=1}^{T} branch_i$
Here, each branch is the reasoning step from the KG which can be defined as Equation 5.

$$branch_i = Subject_i \rightarrow Relation_i \rightarrow Object_i \tag{5}$$

The loss function which is required to optimized during training can be formulated as Equation 6.

$$\mathcal{L}_{\text{KGP}} = -\sum_{t=1}^{m} \log P\Big(y_t \Big| y_{<t}, \texttt{Instruction}, \text{KG}, \texttt{Context}\Big) \tag{6}$$

3. Related Works

The development of multi-hop question generation task relies on the multi-hop datasets where reasoning of multiple evidence are provided. For generating few shot multi-hop question generation, Lin et al. [20] utilized type-aware semantics extraction based chain-of-thoughts prompting. Cao et al. [21] proposed knowledge graph enhanced language model (KGEL) where a GPT-2 model is utilized for encoding the context and answer, whereas an answer-aware GAT with bi-directional attention mechanism provides the path to interact between the answer and the knowledge graph. Finally, Multi-hop questions are generated from the enhanced context representation using multi-head self attention module. Jamshidi and Chali [2] proposed GNET-QG, which integrates a GAT with sequence-to-sequence models. Emerson and Chali [1] proposed a multi-hop question generation approach that applies a transformer model without utilizing any sentence level supporting fact information. MulQG is proposed by Su et al. [22] that is a sequence-to-sequence based model to generate multi-hop questions without any sentence level information. To ensure the complexity of the generated questions, Fei et al. [14] introduced CQG which is a effective controlled framework.

4. Methodology

The proposed methodology of KGP-QG is shown in Figure 1.

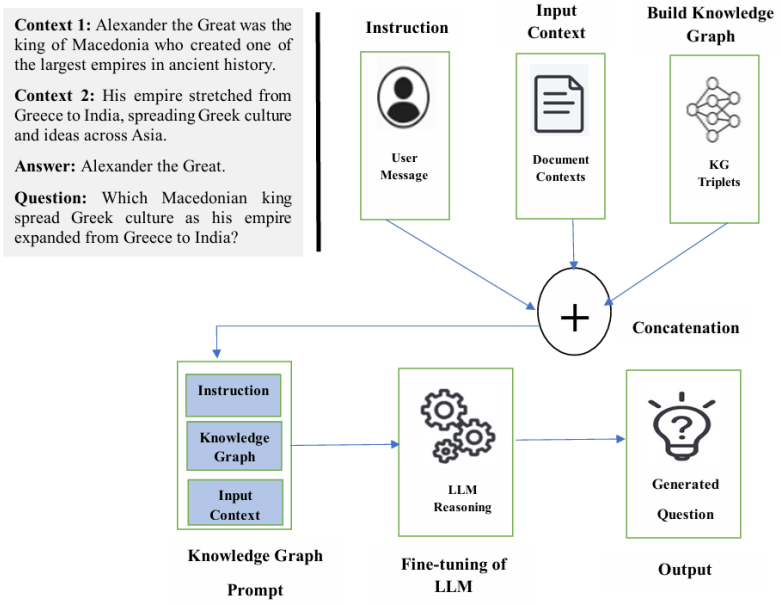

Figure 1: Overview of the KGP-QG Framework.

4.1. Knowledge Graph Generation

To create knowledge graph from input contexts, at first, the annotated text is generated from the input text by performing coreference resolution. Coreference resolution is the task of NLP which identify different expressions in text that refers to the same entity. To find out the relationship between

entities, Stanford CoreNLP [2] toolkit is leveraged. Open Information Extractor (OpenIE) [3] finds the graph structure of the input texts and returns the triplets for each input text.

4.2. Generating Knowledge Graph Prompting

To generate the knowledge graph enhance prompting, instruction, KG, and input texts are fed into the prompts. The instruction describe the task to the LLMs. It also describes how the triplets are formed of the KG. KG is fed as a text prompt and the input text consists the contexts for the multi-hop questions.

Sample Instruction for LLMs

Let's consider the possible branches and generate Multi-hop Questions. Generate reasoning from multiple knowledge graph triplets. Each triplet serves as a branch, and reasoning paths are formed by combining them.

4.3. Encoder

In LLMs, an encoder maps the input sequence into a contextual hidden representation. We have used T5-base [4] and BART-base [5] as encoders separately and have done fine-tuning to update the parameters. The parameters for fine-tuning of the LLMs are shown in Table 1.

Table 1
Specification of parameters to fine-tune the models, both T5-base and BART-base Models for KGP-QG

Parameters	Value
Optimizer	AdamW
Learning rate	1e-4
Warm-up steps	500
Loss function	Cross-entropy
Weight decay	0.01
Patience (early stopping)	3
Schedular	Linear decay
Batch size	8
Train, validation and test split	0.70 : 0.15 :0.15
Epoch	15 (BART-base)
	30 (T5-base)

Let the input sequence $X = [x_1, x_2, \ldots, x_n]$ can be tokenized and embedded as:

$$E_X = [e_1, e_2, \ldots, e_n] \in \mathbb{R}^{n \times d} \tag{7}$$

where,
$e_i = Embedding(x_i) + PositionEmbedding(i)$
d = 768 for BART-base / 512 for T5-base
Multi-head self-attention and feed-forward operations are performed for each encoder layer.

$$H^{(l)} = \text{FFN}\left(\text{MHSA}\left(H^{(l-1)}\right)\right) \tag{8}$$

where,
l: Encoder Layer
MHSA$(\cdot)$: Multi-head self-attention over the full input sequence
FFN$(\cdot)$: Position-wise feed-forward network

[2] https://stanfordnlp.github.io/CoreNLP/.
[3] https://stanfordnlp.github.io/CoreNLP/openIE.html.
[4] https://huggingface.co/google-t5/t5-base.
[5] https://huggingface.co/facebook/bart-base.

4.4. Decoder

By following the transformer architecture, LLMs decoder generates the sequence for multi-hop questions, This process is known as autoregressive generation. Each decoder layer combines two distinct attention mechanism: causal self-attention over past outputs and cross-attention with encoder hidden states.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{9}$$

where,
$Q = XW_Q$: Query for decoder state
$K = XW_K$: Keys
$V = XW_V$: Values
$W_Q, W_K, W_V \in \mathbb{R}^{d \times d_k}$
d_k:Dimensionality of the key vector
X: Decoder hidden states from the previous layers

4.5. Hallucination Detection

To detect the hallucination in the generated multi-hop questions, from semantic similarity methods we have leveraged LaBSE [6] (Language-agnostic BERT Sentence Embedding). It is an embedding based approach which computes the embedding similarity between the reference texts and the generated texts. A lower cosine similarity score indicates a higher probability of hallucination because the generated text diverges semantically from the reference text. In this approach, a threshold value is applied to determine the instances as hallucination or not.

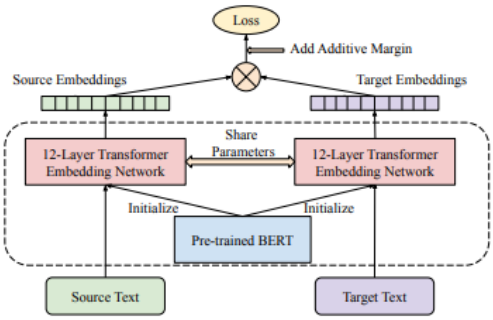

Figure 2: The LaBSE proposed by Feng et al. [23].

LaBSE architecture utilizes a dual encoder based on a shared pre-trained BERT model, one taking the source sentence, other taking in the target sentence [23, 24, 25]. This model consists of 12-layer transformer dual encoders where each encoder is initialized by pre-trained BERT weights [25, 26]. LaBSE tries to maximize of similarity of corresponding generated text and minimize similarity with non-matching sentences. This process is done using the additive margin softmax loss. The probability of a generated text is similar to the reference text and can be represented by Equation 10.

$$P(t_i \mid r_i) = \frac{\exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_i) - m)}{\sum_{j=1}^{N} \exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_j))} \tag{10}$$

where,
r_i: Reference Text
t_i: Generated Text

[6] https://huggingface.co/setu4993/LaBSE.

$\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$:The Resulting Sentence Embedding
sim : The Cosine Similarity
m: Additive Margin
The loss function which is to minimize over batch is the negative log likelihood.

$$L = -\frac{1}{N}\sum_{i=1}^{N} \log P(t_i \mid r_i) \tag{11}$$

5. Experiments

5.1. Dataset

For this research, HotpotQA [27] dataset is utilized. This dataset consists of 113k samples and all the samples are collected from Wikipedia articles. According to Su et al. [22], it is required to remove yes/no answer based data samples to enrich multi-hop ability, we have removed all these types of data samples. We have also removed supporting sentences from the dataset. Also in the HotpotQA dataset, each sample consists of 10 paragraphs where only two paragraphs are relevant to generate the question. We have kept those two paragraphs and discarded eight distractor paragraphs.

5.2. Fine-tuning

To make the pre-trained models task-specific on multi-hop question generation, fine-tuning is done. The fine-tuning is done based on input texts, instructions, and knowledge graphs. That means that, for each input text, knowledge graph is generated, and the instruction, knowledge graph, and text are concatenated for fine-tuning. The HotpotQA dataset is used for this process. In both BART and T5, the models are trained for 50 epochs, where early stopping is used as a regularization technique to prevent the model from overfitting. In this work, 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 30 epochs, whereas for the BART-base model, the best outcome is achieved after 15 epochs. The Adam optimizer [28] is used, where the Cross-Entropy function [29] is used to calculate the loss. The learning rate is fixed as 10^{-4}. The whole configuration is shown in Table 1.

5.3. Experimental Setup

The main computational resources of this research are Narval and Cedar servers from Compute Canada [7]. Narval server provided A100 GPUs, and Cedar provides V100 GPUs. We have used the Distributed Data Parallel (DDP) [8] feature in PyTorch to utilize 4 GPUs simultaneously. Details, code and implementations are available on our Github [9] repository.

6. Evaluation

To evaluate the performance of our model, we have compared the generated texts with the reference texts by utilizing automated evaluation metrics. Based on the widespread utilization in QG tasks, we have used BLEU [30], ROUGE [31], and METEOR [32] as evaluation metrics. After fine-tuning the model using the T5-base and BART-base, we have compared the results with the existing methodologies of MHQG, which is shown in Table 2. The methodologies are chosen based on the Prompting approach (TASE-CoT by Lin et al. [20]), the Graph-based approach (KGEL by Cao et al. [21], GNET-QG by Jamshidi

[7] https://docs.alliancecan.ca/.
[8] https://docs.alliancecan.ca/wiki/PyTorch.
[9] https://github.com/AlHasibMahamud/KGP-QG_Multihop-Question-Generation-based-on-Knowledge-Graph-Based-Prompting.

and Chali [2], MulQG by Su et al. [22], DCQG by Cheng et al. [33]), and the Transformer-based approach by Emerson and Chali [1].

Table 2
Performance Comparison between KGP-QG and Existing Models

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
CQG	49.71	37.04	29.93	25.09	41.83	27.45
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**
KGEL	41.93	28.04	20.83	16.13	19.70	35.28
Transformer Based	42.13	30.44	23.84	19.42	39.26	22.78
MultiFactor	54.17	41.50	33.74	28.22	28.60	44.17
TASE-CoT	45.89	34.06	27.11	22.37	39.68	23.39
KGP-QG(T5 Backbone)	51.12	41.35	36.79	33.35	41.27	42.93
KGP-QG(BART Backbone)	**58.02**	**49.00**	**44.49**	**40.87**	**49.55**	49.84

From Table 2, we can see that our model KGP-QG has outperformed across all the metrics except METEOR, where the difference between our model KGP-QG and GNET-QG is only 0.03.
During the fine-tuning of LLMs, adding instructions in prompts improves output metrics as instructions guide the models explicitly to understand the task requirements, perform the task, and align their response accordingly. Adding KG with instructions and input texts guides the LLMs to think step-by-step, provides the path for LLMs to understand the subject, relationship, and object from the input texts, and enriches the output involving complex reasoning.

6.1. Human Evaluation

We have performed a human evaluation comparing the generated texts and the reference texts. We have selected 300 samples from the test set and three native English annotators marked each sample from 1 (very poor) to 5 (very good) based on the following criteria, except for multi-hop relevance. A Fleiss' Kappa [34] score of 0.60 is achieved between three annotators, indicating a fair degree of consistency in their assesments.

- **Fluency:** Grammatical correctness and readability
- **Answerability:** Based on input texts, the generated question is answerable or not
- **Completeness:** Fully formed question or not
- **Multi-hop Relevance:** Multi-hop reasoning of the generated questions

Table 3
Human Evaluation Scores

Model	Fluency	Completeness	Answerability	Multi-hop Relevance
BART	3.86	3.96	3.97	54.0%
GNET-QG	3.94	4.14	4.18	76.0%
KGP-QG	4.30	4.28	4.50	82.0%
Human	4.38	4.42	4.56	85.50%

Further, we average the scores on each question, and the result is shown in Table 3. Table 3 also shows the comparison of human evaluation by comparing with the human evaluation results of BART and GNET-QG.

6.2. Hallucination Detection

To find the hallucination from the generated multi-hop questions, we have calculated the cosine similarities from each pair of reference questions and the generated question after encoding each pair

using the LaBSE model. As the threshold value is chosen to be 0.5, if the similarity score of the pair is less than 0.5, then the generated question from that pair is marked as hallucination. We have calculated the total number of hallucinated pairs and the percentage of likely hallucinated pairs, which is shown in Table 4.

Table 4
Quantitative Analysis of Hallucinations

Model Name	Threshold	Total # of Pairs	# of Likely Hallucinated Pairs	% of Likely Hallucinated Pairs
KGP-QG (T5-backbone)	0.5	13788	2332	16.91%
KGP-QG (BART-backbone)	0.5	13788	1948	14.13%

From Table 4, it is clear that the number of hallucinated multi-hop questions is very low. Among 13788 generated multi-hop questions, the number of hallucinations for the T5-Base and BART-Base models as backbones is 2332 and 1948, respectively, accounting for 16.91% and 14.13%. So, we can say that the BART-Base model, as the backbone of the KGP-QG framework, is more suitable for reducing hallucination.

7. Conclusion

We proposed the KGP-QG framework that combines instruction, knowledge graph, and input context with an LLM to ensure LLM reasoning based on the knowledge graph. Automatic evaluation metrics and human evaluation show the effectiveness of KGP-QG, which offers a valuable baseline for future research. The integration of knowledge graph prompting with LLMs is the major contribution of this research. Our approach has also reduced hallucinations in the generated multi-hop questions. Further, this approach can be utilized in other natural language generation tasks like low-resource question generation, reasoning in multi-modal question-answering, task-specific text generation, etc.

Declaration of Generative AI Use

The authors declare that no generative AI tools or AI-assisted technologies were used in the research, analysis, or writing process of this manuscript. All content, including analysis and writing, was produced solely by the authors.

References

[1] J. Emerson, Y. Chali, Multi-hop question generation without supporting fact information, The International FLAIRS Conference Proceedings 36 (2023). URL: https://journals.flvc.org/FLAIRS/article/view/133320. doi:10.32473/flairs.36.133320.

[2] S. Jamshidi, Y. Chali, Gnet-qg: Graph network for multi-hop question generation, COLING 2025 (2025) 20.

[3] W. Huang, G. Zhou, M. Lapata, P. Vougiouklis, S. Montella, J. Z. Pan, Prompting large language models with knowledge graphs for question answering involving long-tail facts, 2024. URL: https://arxiv.org/abs/2405.06524. arXiv:2405.06524.

[4] Y. Wang, N. Lipka, R. A. Rossi, A. Siu, R. Zhang, T. Derr, Knowledge graph prompting for multi-document question answering, 2023. URL: https://arxiv.org/abs/2308.11730. arXiv:2308.11730.

[5] D. Chen, A. Fisch, J. Weston, A. Bordes, Reading wikipedia to answer open-domain questions, 2017. URL: https://arxiv.org/abs/1704.00051. arXiv:1704.00051.

[6] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: B. Webber, T. Cohn, Y. He, Y. Liu

(Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6769–6781. URL: https://aclanthology.org/2020.emnlp-main.550/. doi:10.18653/v1/2020.emnlp-main.550.

[7] H. P. Zou, C. Caragea, JointMatch: A unified approach for diverse and collaborative pseudo-labeling to semi-supervised text classification, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 7290–7301. URL: https://aclanthology.org/2023.emnlp-main.451/. doi:10.18653/v1/2023.emnlp-main.451.

[8] J. Dong, Q. Zhang, X. Huang, K. Duan, Q. Tan, Z. Jiang, Hierarchy-aware multi-hop question answering over knowledge graphs, 2023, pp. 2519–2527. doi:10.1145/3543507.3583376.

[9] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, K. Narasimhan, Tree of thoughts: Deliberate problem solving with large language models, 2023. URL: https://arxiv.org/abs/2305.10601. arXiv:2305.10601.

[10] H. Trivedi, N. Balasubramanian, T. Khot, A. Sabharwal, Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions, 2023. URL: https://arxiv.org/abs/2212.10509. arXiv:2212.10509.

[11] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, 2023. URL: https://arxiv.org/abs/2201.11903. arXiv:2201.11903.

[12] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk, T. Hoefler, Graph of thoughts: Solving elaborate problems with large language models, Proceedings of the AAAI Conference on Artificial Intelligence 38 (2024) 17682–17690. URL: http://dx.doi.org/10.1609/aaai.v38i16.29720. doi:10.1609/aaai.v38i16.29720.

[13] M. Pietrasik, M. Reformat, A. Wilbik, Hierarchical blockmodelling for knowledge graphs, 2024. URL: https://arxiv.org/abs/2408.15649. arXiv:2408.15649.

[14] Z. Fei, Q. Zhang, T. Gui, D. Liang, S. Wang, W. Wu, X. Huang, CQG: A simple and effective controlled generation framework for multi-hop question generation, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 6896–6906. URL: https://aclanthology.org/2022.acl-long.475/. doi:10.18653/v1/2022.acl-long.475.

[15] Y. Tang, T. Liu, G. Liu, J. Li, R. Dai, C. Yuan, Enhancement of power equipment management using knowledge graph, 2019. URL: https://arxiv.org/abs/1904.12242. arXiv:1904.12242.

[16] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Computing Surveys 54 (2021) 1–37. URL: http://dx.doi.org/10.1145/3447772. doi:10.1145/3447772.

[17] A. Alansari, H. Luqman, Large language models hallucination: A comprehensive survey, 2025. URL: https://arxiv.org/abs/2510.06265. arXiv:2510.06265.

[18] J. Maynez, S. Narayan, B. Bohnet, R. McDonald, On faithfulness and factuality in abstractive summarization, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 1906–1919. URL: https://aclanthology.org/2020.acl-main.173/. doi:10.18653/v1/2020.acl-main.173.

[19] D. Anh-Hoang, V. Tran, L.-M. Nguyen, Survey and analysis of hallucinations in large language models: attribution to prompting strategies or model behavior, Frontiers in Artificial Intelligence Volume 8 - 2025 (2025). URL: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1622292. doi:10.3389/frai.2025.1622292.

[20] Z. Lin, W. Chen, Y. Song, Y. Zhang, Prompting few-shot multi-hop question generation via comprehending type-aware semantics, in: K. Duh, H. Gomez, S. Bethard (Eds.), Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics,

Mexico City, Mexico, 2024, pp. 3730–3740. URL: https://aclanthology.org/2024.findings-naacl.236/. doi:10.18653/v1/2024.findings-naacl.236.

[21] Z. Cao, P. Li, Y. Zhong, S. Li, Multi-hop question generation with knowledge graph-enhanced language model, Applied Sciences 13 (2023). doi:10.3390/app13095765.

[22] D. Su, Y. Xu, W. Dai, Z. Ji, T. Yu, P. Fung, Multi-hop question generation with graph convolutional network, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 4636–4647. URL: https://aclanthology.org/2020.findings-emnlp.416/. doi:10.18653/v1/2020.findings-emnlp.416.

[23] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, W. Wang, Language-agnostic BERT sentence embedding, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 878–891. URL: https://aclanthology.org/2022.acl-long.62/. doi:10.18653/v1/2022.acl-long.62.

[24] Y. Yang, G. H. Abrego, S. Yuan, M. Guo, Q. Shen, D. Cer, Y. hsuan Sung, B. Strope, R. Kurzweil, Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax, 2019. URL: https://arxiv.org/abs/1902.08564. arXiv:1902.08564.

[25] E. A. Chimoto, B. A. Bassett, Very low resource sentence alignment: Luhya and Swahili, in: A. K. Ojha, C.-H. Liu, E. Vylomova, J. Abbott, J. Washington, N. Oco, T. A. Pirinen, V. Malykh, V. Logacheva, X. Zhao (Eds.), Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022), Association for Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 1–8. URL: https://aclanthology.org/2022.loresmt-1.1/.

[26] M. Guo, Q. Shen, Y. Yang, H. Ge, D. Cer, G. Hernandez Abrego, K. Stevens, N. Constant, Y.-H. Sung, B. Strope, R. Kurzweil, Effective parallel corpus mining using bilingual sentence embeddings, in: O. Bojar, R. Chatterjee, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, C. Monz, M. Negri, A. Névéol, M. Neves, M. Post, L. Specia, M. Turchi, K. Verspoor (Eds.), Proceedings of the Third Conference on Machine Translation: Research Papers, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 165–176. URL: https://aclanthology.org/W18-6317/. doi:10.18653/v1/W18-6317.

[27] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, C. D. Manning, HotpotQA: A dataset for diverse, explainable multi-hop question answering, in: E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2369–2380. URL: https://aclanthology.org/D18-1259/. doi:10.18653/v1/D18-1259.

[28] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. URL: https://arxiv.org/abs/1412.6980. arXiv:1412.6980.

[29] A. Mao, M. Mohri, Y. Zhong, Cross-entropy loss functions: Theoretical analysis and applications, 2023. URL: https://arxiv.org/abs/2304.07288. arXiv:2304.07288.

[30] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: P. Isabelle, E. Charniak, D. Lin (Eds.), Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. URL: https://aclanthology.org/P02-1040/. doi:10.3115/1073083.1073135.

[31] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013/.

[32] A. Lavie, A. Agarwal, METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments, in: C. Callison-Burch, P. Koehn, C. S. Fordyce, C. Monz (Eds.), Proceedings of the Second Workshop on Statistical Machine Translation, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 228–231. URL: https://aclanthology.org/W07-0734/.

[33] Y. Cheng, S. Li, B. Liu, R. Zhao, S. Li, C. Lin, Y. Zheng, Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.),

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 5968–5978. URL: https://aclanthology.org/2021.acl-long.465/. doi:10.18653/v1/2021.acl-long.465.

[34] J. Fleiss, Measuring nominal scale agreement among many raters, Psychological Bulletin 76 (1971) 378–382. doi:10.1037/h0031619.

A. Example of Model Prediction

Example 1
Predicted Question: What is the domestic football capacity of the stadium where The 2011 Conference Premier Football Final was held?
Reference Question: What is the domestic football capacity of the stadium which hosted the 2011 Conference Premier play-off Final?
Comparison: Both questions are looking for the same information. The reference question is grammatically slightly more correct.

Example 2
Predicted Question:Bendy Casimir made his WEC debut against which American mixed martial artist who currently fights as a Featherweight in the Ultimate Fighting Championship organization?
Reference Question: Bendy Casimir made his WEC debut against which American MMA fighter?
Comparison: Both of the questions are seeking the same information. The predicted question is more precise.

Example 3
Predicted Question: What experimental work is the French physicist who is on the selection committee for the John Stewart Bell Prize known for?
Reference Question: What experimental work is the French physicist, born in 1947 who serves on the selection committee for the John Stewart Bell Prize established in 2009, well-known for?
Comparison: Both the questions are almost identical.

Example 4
Predicted Question: Where is the headquarters of the team that has a Swiss professional racing driver as a reserve driver?
Reference Question: Where is the headquarters of the team that includes a Swiss professional racing driver born in 1988?
Comparison: The reference question is better as it provides more information.

Example 5
Predicted Question: Michael Jones and Rida Johnson Young, have which mutual occupation?
Reference Question: James Young and Michael Jones, share which common occupations?
Comparison: Both the questions are asking same information.

Example 6
Predicted Question: One of the longest-running Bourbon on the market today was produced by what company?
Reference Question: Who is the founder of the company that markets Jack Daniel's, and Finlandia Vodka?
Comparison: The Generated Question is asking separate information from the Reference Question. Though both of the questions are input-related multi-hop questions, the Generated Question is better as it creates more ties with both input paragraphs.

B. Knowledge Graph Prompting

Listing 1: The Prompt for Generating Multi-hop Question for KGPQG

```
###Instruction
Let's consider the possible branches and generate Multi-hop Questions. Generate
    reasoning from multiple knowledge graph triplets. Each triplet serves as a
    branch, and reasoning paths are formed by combining them.
{Knowledge Graph}
###Example
input: {Paragraph 1}
       {Paragraph 2}
output: {Multi-hop Question}
Now Generate Multi-hop Question for the following example.
input: {Paragraph 1}
       {Paragraph 2}
output:
```