GNET-QG: Graph Network for Multi-hop Question Generation

Samin Jamshidi
University of Lethbridge
Department of Computer Science
jamshidisamin73@gmail.com

Yllias Chali
University of Lethbridge
Department of Computer Science
yllias.chali@uleth.ca

Abstract

Multi-hop question generation is a challenging task in natural language processing (NLP) that requires synthesizing information from multiple sources. We propose GNET-QG, a novel approach that integrates Graph Attention Networks (GAT) with sequence-to-sequence models, enabling structured reasoning over multiple information sources to generate complex questions. Our experiments demonstrate that GNET-QG outperforms previous state-of-the-art models across several evaluation metrics, particularly excelling in METEOR, showing its effectiveness in enhancing machine reasoning capabilities.

1 Introduction

Question generation (QG) is the task of producing a natural language question given an input context and an answer. While recent neural models have achieved considerable success in QG, they often fail to generate complex, multi-hop questions that require reasoning across multiple contexts. Unlike simple questions that rely on a single fact, multi-hop questions demand the integration of knowledge from multiple pieces of information to formulate a coherent query.

Multi-hop question generation is not only a challenging task but also a critical one, with applications ranging from improving query suggestions for search engines to enhancing educational tools for reading comprehension (Zamani et al., 2020; Heilman and Smith, 2010) . Despite advancements in models like MulQG (Su et al., 2020) and CQG (Fei et al., 2022), existing methods still struggle to consistently generate high-quality multi-hop questions across diverse contexts.

To address this gap, we introduce GNET-QG, a model that incorporates Graph Attention Networks (GAT) to identify and focus on relevant entities within a context. By enriching the input context using GAT and combining it with a powerful sequence-to-sequence model, GNET-QG is able to generate more complex, coherent, and answerable questions. Our method shows significant improvements over existing techniques, especially in the METEOR metric, indicating better semantic alignment in generated questions.

Context1:
The Oberoi family is an Indian family that is famous for its involvement in hotels, namely through The Oberoi Group .

Context2:
The Oberoi Group is a hotel company with its head office in Delhi .

Answer: Delhi

Simple Question: Where is the head office of The Oberoi Group ?

Complex Question: Oberoi family is part of a hotel company that has a head office in what city?

Figure 1: Example of multi-hop question generation. Context 1 introduces the Oberoi family and their connection to The Oberoi Group, while Context 2 provides information about the group's head office location. A simple question asks for the head office location directly, while a multi-hop question integrates information from both contexts to identify the head office city. In this context, 'complexity' refers to the need for synthesizing information from multiple contexts. This example is adapted from the HotpotQA dataset (Yang et al., 2018).

2 Related Work

Early research in question generation focused on rule-based methods for single-hop QG, transforming declarative sentences into questions via handcrafted rules (Heilman and Smith, 2010). Neu-

Proceedings of the Generative AI and Knowledge Graph Workshop (GenAIK), pages 20–26
January 19, 2025.

ral methods soon emerged, leveraging sequence-to-sequence models to improve reading comprehension QG, but struggled with multi-hop reasoning due to their reliance on local context (Du and Cardie, 2017).

Recent approaches, such as MulQG (Su et al., 2020) and CQG (Fei et al., 2022), have incorporated graph-based models to address multi-hop QG. MulQG used Graph Convolutional Networks (GCNs) to represent context and perform multi-hop encoding fusion, while CQG utilized a controlled framework to focus on key entities during question generation. Building upon these methods, Lin et al. (2024) introduced a type-aware semantics extraction-based chain-of-thought (TASE-CoT) approach for few-shot multi-hop QG. This approach begins by identifying question types and key semantic phrases from the provided documents and answer, then utilizes a three-step chain-of-thought template to generate multi-hop questions based on the extracted information. Despite these advancements, there remains a need for models that consistently perform well across different datasets and contexts.

GNET-QG builds on this body of work by integrating Graph Attention Networks (GAT) into the question generation process. GAT enables our model to focus attention on important entities within the input context, providing better entity representation for complex reasoning tasks.

3 Research Methodology

3.1 Motivation and Overview

Multi-hop question generation (QG) requires reasoning across multiple interconnected information pieces in a document. Traditional transformer models perform well in single-hop QG but struggle with multi-hop tasks due to limitations in capturing long-range dependencies and complex entity relationships. Existing methods often overlook explicit modeling of these relationships, leading to less coherent questions that lack true multi-hop reasoning.

Moreover, many current approaches are tightly coupled with specific models, lacking the flexibility to adapt to newer or different large language models (LLMs). This rigidity hinders leveraging advancements in the field and limits applicability across diverse architectures.

To address these limitations, we propose **GNET-QG**, a model integrating a graph entity network with a transformer-based architecture. Our approach enhances the quality and complexity of generated questions by explicitly modeling semantic relationships between entities through an entity graph and enriching the input context for the question generation model. Crucially, the architecture is designed to be compatible with various transformer-based models due to its text-based enriched input context, allowing greater flexibility and adaptability.

In our experiments, GNET-QG demonstrates significant improvements over baseline models in generating coherent and complex multi-hop questions. We successfully integrate both BART and T5 models within our architecture, evidencing its compatibility and effectiveness with different LLMs.

3.2 Constructing the Entity Graph

To capture the relationships essential for multi-hop reasoning, we construct an entity graph where nodes represent entities extracted from the document, and edges represent relationships between these entities. Entities are identified using a BERT-based Named Entity Recognition (NER) model, resulting in a set $E = \{e_0, e_1, \ldots, e_n\}$, Entities and relationships are derived from the preprocessed data provided by MULQG (Su et al., 2020).

Edges between nodes are defined as follows:

- **Same Sentence Co-occurrence**: Nodes are connected if they appear within the same sentence, capturing immediate contextual relationships.
- **Paragraph Title Relations**: Nodes are connected if a paragraph's title contains an entity that also appears within the paragraph, highlighting hierarchical and topical associations.
- **Cross-Paragraph Entity Consistency**: Entities appearing in different paragraphs but referring to the same concept are linked, ensuring consistency across the document.

3.3 Enriching the Input Context

To effectively leverage the constructed entity graph, we use a Graph Attention Network (GAT) to enhance node representations. The GAT computes attention scores and updates node features as follows:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}_i^T [Wh_i \parallel Wh_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(\mathbf{a}_i^T [Wh_i \parallel Wh_k]))} \tag{1}$$

$$h_i' = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j\right) \tag{2}$$

In these equations, h_i and h_j represent the feature vectors of nodes i and j, respectively. $\mathcal{N}(i)$ denotes the set of neighbors of entity i, while W is a weight matrix that transforms input features into a new space. The updated representation h_i' for node i is computed by aggregating information from its neighbors. The attention coefficient α_{ij} represents the importance of node j's features in updating node i. The learnable vector $\mathbf{a}_i$ projects the concatenated feature vector $[Wh_i \parallel Wh_j]$ into a scalar score, allowing the model to compute the relevance of neighboring nodes.

To further enhance the contextual information, we apply multi-head attention at each step. This mechanism allows the model to capture different aspects of the neighbors' features across multiple attention "heads." Multi-head attention is defined as follows:

$$\text{MultiHead}(h_i) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h), \tag{3}$$

where each head is computed as:

$$\text{head}_k = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(k)} W^{(k)} h_j\right). \tag{4}$$

Here, h denotes the number of attention heads, and $W^{(k)}$ is the weight matrix for the k-th head, and $\alpha_{ij}^{(k)}$ is the attention coefficient for the k-th head. This multi-head setup allows the GAT to capture a richer representation by focusing on different aspects of the input.

After applying the multi-head GAT, we flatten the updated node features H' and pass them through a linear transformation followed by a sigmoid activation function:

$$H_{\text{flat}} = \text{Flatten}(H') \tag{5}$$

$$H_{\text{linear}} = W_l H_{\text{flat}} + b_l \tag{6}$$

$$P = \sigma(H_{\text{linear}}) \tag{7}$$

In these equations, W_l and b_l are learnable parameters of the linear layer, and σ represents the sigmoid activation function, which outputs probability scores for each node. Nodes with probability scores greater than 0.5 are selected:

$$E_{\text{sub}} = \{h_i' \in H' \mid P_i > 0.5\} \tag{8}$$

Here, the selected nodes E_{sub} are the textual representations of entities. These textual representations of entities are concatenated with the original context C and the answer A to form the "enriched input context":

$$C_{\text{enriched}} = [C; A; E_{\text{sub}}] \tag{9}$$

The enriched input context, now in textual form, is subsequently fed into the transformer encoder for question generation, enhancing the model's ability to integrate multi-hop reasoning with focused entity relationships.

3.4 Encoder-Decoder Framework

The enriched input context is fed into the encoder of a pre-trained transformer model, such as BART or T5. The encoder generates contextualized embeddings that provide a compact representation of the input. Incorporating the enriched context enhances the encoder's ability to process long-range dependencies and relationships.

The decoder generates the output question autoregressively, utilizing the encoder's embeddings. It applies masked self-attention to ensure each token prediction considers only previously generated tokens, producing coherent and contextually appropriate multi-hop questions.

Figure 2 illustrates the encoder component of GNET-QG with a BART backbone. Initially, entities (nodes) from the contexts (C) are identified and labeled as E_0. A graph is created using these entities, capturing their relationships based on co-occurrence and paragraph structure. The entity graph is then passed to the Graph Attention Network (GAT), which processes the entity features. After applying flattening, a linear transformation, and a sigmoid activation, the resulting entity representations are concatenated with the contexts to form the enriched input context. This enriched input is subsequently fed into the BART encoder.

4 Implementation Details

In the task of multi-hop question generation, we implemented **GNET-QG** by integrating the GAT from this GitHub repository[1] with an encoder-decoder

[1] https://github.com/HLTCHKUST/MulQG

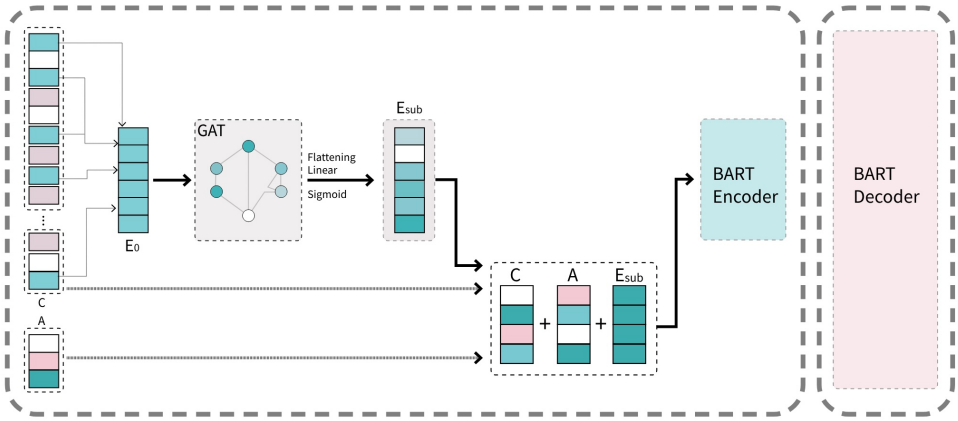

Figure 2: Full architecture of GNET-QG with BART backbone. C represents the original context extracted from the input, A denotes the answer provided as input to the model, and E_{sub} consists of the selected textual entities derived from the input context using the Graph Attention Network (GAT). These components are concatenated to form the enriched input context fed into the BART encoder.

model as the main backbone. We connected the untrained GAT architecture with pre-trained versions of both BART and T5 models to generate the multi-hop questions, leveraging the filtered HotpotQA dataset, following the preprocessing approach outlined in the MULQG (Su et al., 2020).

The GAT and the transformer models (BART and T5) were trained **end-to-end**. During training, gradients were backpropagated through both the GAT and the transformer model, allowing the entire network to learn jointly. This approach enables the model to effectively incorporate the structural information captured by the GAT into the question generation process.

Our candidate pre-trained models for the proposed architecture were the `bart-squad-qg-hl`[2] version of the BART model and the `t5-base-finetuned-question-generation-ap`[3] version of the T5 model. These models were selected due to their proven efficacy in question generation tasks and their availability for fine-tuning on domain-specific datasets.

4.1 Automatic Evaluation

To evaluate GNET-QG's effectiveness, we employed automated evaluation metrics to assess its predictions on samples from the test dataset, comparing the generated questions from the model with the reference questions from the dataset. The metrics used include BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), and METEOR (Lavie and Agarwal, 2007), chosen for their extensive adoption in the question-generation research field. This evaluation allows us to directly compare GNET-QG's performance with previous studies on multi-hop question generation.

In Table 1, we present comparative results of GNET-QG with BART backbones against other models, including BART by Lewis et al. (2019), MulQG by Su et al. (2020), CQG by Fei et al. (2022), and TASE-CoT by Lin et al. (2024).

Our model demonstrates superior performance, particularly in the METEOR metric. This improvement can be attributed to GNET-QG's ability to generate questions that are semantically richer and more closely aligned with the reference questions. METEOR places greater emphasis on semantic similarity, synonymy, and recall, rewarding models that capture the meaning of the reference even if the exact wording differs. By explicitly modeling semantic relationships between entities and enriching the input context through the graph entity network, GNET-QG generates questions that include relevant synonyms, paraphrases, and morphological variants, all of which METEOR recognizes and rewards. This focus on semantic richness and relevance allows our model to outperform others in METEOR, highlighting its effectiveness in producing high-quality, semantically accurate multi-hop questions.

Furthermore, the enriched input context enables our model to capture more of the necessary information required to formulate comprehensive questions, increasing recall, a component heavily weighted in METEOR. In contrast, other models

[2] https://github.com/p208p2002/Transformer-QG-on-SQuAD

[3] https://github.com/patil-suraj/question_generation

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
CQG	49.71	37.04	29.93	25.09	**41.83**	27.45
BART	41.41	30.90	24.39	19.75	36.13	25.20
TASE-CoT	45.89	34.06	27.11	22.37	39.68	23.39
GNET-QG (BART backbone)	**49.72**	**38.95**	**32.88**	**27.93**	40.25	**49.87**

Table 1: Automatic evaluation results on HotpotQA. The table compares the performance of various models across multiple metrics, including BLEU, ROUGE-L, and METEOR.

may excel in n-gram precision metrics like BLEU but may not capture the deeper semantic nuances that METEOR evaluates. Therefore, GNET-QG's superior performance in METEOR underscores its capability to generate questions that are not only grammatically correct but also semantically meaningful and contextually appropriate.

4.2 Human Evaluation

To comprehensively assess the performance of our model, we performed a human evaluation comparing the questions generated from three sources: the baseline BART model, our proposed GNET-QG model and human-generated questions. We evaluated the questions based on four metrics:

- **Fluency**: Grammatical correctness and readability.
- **Completeness**: Whether the questions are fully formed and coherent.
- **Answerability**: If the questions are answerable based on the given context.
- **Multi-hop Relevance**: Whether the questions require synthesizing information from multiple contexts (binary classification).

Each metric, except for Multi-hop Relevance, was rated on a scale from 1 to 5, with higher scores indicating better performance. Five annotators evaluated 50 randomly sampled test cases from the test set.

The results are summarized in Tables 2 and 3. GNET-QG achieved a 76% rate of generating multi-hop questions, significantly higher than BART's 54%. In terms of question quality, GNET-QG scored higher in Completeness (4.14) and Answerability (4.18) compared to BART's scores of 3.96 and 3.97, respectively. Although GNET-QG showed improvements in fluency over BART, it still fell slightly short of human-generated questions, suggesting room for further refinement.

Models	Yes	No	Percentage (% Yes)
BART	27	23	54.0
GNET-QG	**38**	**12**	**76.0**
Human	40	10	80.0

Table 2: Counts and percentages of multi-hop questions generated by each model.

Models	Completeness	Answerability	Fluency
BART	3.96	3.97	3.86
GNET-QG	**4.14**	**4.18**	**3.94**
Human	4.28	4.42	4.30

Table 3: Mean ratings for Completeness, Answerability, and Fluency.

4.3 Model Compatibility and Experimental Evidence

A key advantage of our GNET-QG architecture is its compatibility with various large language models (LLMs). Since the enriched input context is text-based, it integrates seamlessly with any transformer-based model capable of processing text input. To demonstrate this flexibility, we implemented GNET-QG using both BART and T5 backbones and compared the results against standard fine-tuned versions of BART and T5.

For the baseline models, we utilized BART as proposed by Lewis et al. (2019) and the fine-tuned version of T5, specifically `t5-base-finetuned-question-generation-ap`[4]. We evaluated the models on the HotpotQA dataset to ensure a fair comparison. The results are presented in Table 4, which shows the superior performance of GNET-QG, especially in terms of METEOR scores, in both the BART and T5 backbones.

These results highlight the architecture's ability to enhance performance across different transformer models, validating its effectiveness and flex-

[4] https://github.com/patil-suraj/question_generation

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
BART Baseline	41.41	30.90	24.39	19.75	36.13	25.20
GNET-QG (BART backbone)	**49.72**	**38.95**	**32.88**	**27.93**	**40.25**	**49.87**
T5 Baseline	32.08	22.04	17.26	13.68	30.78	27.72
GNET-QG (T5 backbone)	42.10	31.92	26.42	22.05	34.38	42.51

Table 4: Comparison of performance metrics (BLEU-1 to BLEU-4, ROUGE-L, and METEOR) for question generation on HotpotQA dataset. GNET-QG shows significant improvements with both BART and T5 backbones.

ibility.

5 Conclusion

In this work, we introduced **GNET-QG**, a graph-based approach to multi-hop question generation that effectively reduces model complexity without sacrificing performance. By explicitly modeling semantic relationships between entities and enriching the input context for transformer-based models, GNET-QG addresses the limitations of existing methods. A key contribution of GNET-QG is its ability to reduce the model size by approximately 7.5 million parameters compared to CQG, a highly competitive model in this space. This substantial reduction leads to improvements in computational efficiency, lower memory usage, and faster inference speeds, making GNET-QG a more practical and scalable solution for real-world applications. Despite the smaller parameter size, our experimental results demonstrate that GNET-QG outperforms CQG in terms of the quality of generated questions, highlighting its effectiveness and efficiency.

Furthermore, we validated the versatility of our architecture by integrating well-known sequence-to-sequence frameworks such as BART and T5. The consistent performance improvements across these models underscore GNET-QG's compatibility with different large language models and its ability to generate high-quality, complex multi-hop questions requiring sophisticated reasoning over multiple interconnected pieces of information.

Future research could focus on enhancing the model's reasoning capabilities to better address abstract or causal questions and extending its application to other NLP tasks like summarization or machine translation. Additionally, exploring GNET-QG's performance on non-English datasets could unlock its potential for multilingual question generation, further broadening its impact.

References

Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2067–2073.

Zichu Fei, Qi Zhang, Tao Gui, Di Liang, Sirui Wang, Wei Wu, and Xuan-Jing Huang. 2022. Cqg: A simple and effective controlled generation framework for multi-hop question generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6896–6906.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zefeng Lin, Weidong Chen, Yan Song, and Yongdong Zhang. 2024. Prompting few-shot multi-hop question generation via comprehending type-aware semantics. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3730–3740.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Dan Su, Yan Xu, Wenliang Dai, Ziwei Ji, Tiezheng Yu, and Pascale Fung. 2020. Multi-hop question generation with graph convolutional network. *arXiv preprint arXiv:2010.09240*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating clarifying questions for information retrieval. In *Proceedings of the web conference 2020*, pages 418–428.