Multi-hop Question Generation with Graph Convolutional Network

Dan Su, Yan Xu, Wenliang Dai, Ziwei Ji, Tiezheng Yu, Pascale Fung
Center for Artificial Intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
{dsu, yxucb, wdaiai, zjiad, tyuah}@connect.ust.hk,
pascale@ece.ust.hk

Abstract

Multi-hop Question Generation (QG) aims to generate answer-related questions by *aggregating* and *reasoning* over multiple scattered evidence from different paragraphs. It is a more challenging yet under-explored task compared to conventional single-hop QG, where the questions are generated from the sentence containing the answer or nearby sentences in the same paragraph without complex reasoning. To address the additional challenges in multi-hop QG, we propose Multi-Hop Encoding Fusion Network for Question Generation (MulQG), which does context encoding in multiple hops with Graph Convolutional Network and encoding fusion via an Encoder Reasoning Gate. To the best of our knowledge, we are the first to tackle the challenge of multi-hop reasoning over paragraphs without any sentence-level information. Empirical results on HotpotQA dataset demonstrate the effectiveness of our method, in comparison with baselines on automatic evaluation metrics. Moreover, from the human evaluation, our proposed model is able to generate fluent questions with high completeness and outperforms the strongest baseline by 20.8% in the multi-hop evaluation. The code is publicly available at https://github.com/HLTCHKUST/MulQG.

1 Introduction

Question Generation (QG) is a task to automatically generate a question from a given context and, optionally, an answer. Recently, we have observed an increasing interest in text-based QG (Du et al., 2017; Zhao et al., 2018; Scialom et al., 2019; Nema et al., 2019; Zhang and Bansal, 2019).

Most of the existing works on text-based QG focus on generating SQuAD-style (Rajpurkar et al., 2016; Puri et al., 2020) questions, which are generated from the sentence containing the answer or nearby sentences in the same paragraph, via single-hop reasoning (Zhou et al., 2017; Zhao et al., 2018). Little effort has been put in multi-hop QG, which is a more challenging task. Multi-hop QG requires *aggregating* several scattered evidence spans from multiple paragraphs, and *reasoning* over them to generate answer-related, factual-coherent questions. It can serve as an essential component in education systems (Heilman and Smith, 2010; Lindberg et al., 2013; Yao et al., 2018), or be applied in intelligent virtual assistant systems (Shum et al., 2018; Pan et al., 2019). It can also combine with question answering (QA) models as dual tasks to boost QA systems with reasoning ability (Tang et al., 2017).

Paragraph A: Marine Tactical Air Command Squadron 28 (*Location T*) is a United States Marine Corps aviation command and control unit based at Marine Corps Air Station Cherry Point (*Location C*) ...
Paragraph B: Marine Corps Air Station Cherry Point (*Location C*) ... is a United States Marine Corps airfield located in Havelock, North Carolina (*Location H*), USA ...
Answer: Havelock, North Carolina (*Location H*)
Question: What city is the Marine Air Control Group 28 (*Location T*) located in?

Table 1: An example of multi-hop QG in the HotpotQA (Yang et al., 2018) dataset. Given the answer is Location H, to ask where is T located, the model needs a bridging evidence to know that T is located in C, and C is located in H ($T \rightarrow C \rightarrow H$). This is done by multi-hop reasoning.

Intuitively, there are two main additional challenges needed to be addressed for multi-hop QG. The first challenge is how to effectively identify scattered pieces of evidence that can connect the reasoning path of the answer and question (Chauhan et al., 2020). As the example shown

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4636–4647

in Table 1, to generate a question asking about *"Marine Air Control Group 28"* given only the answer *"Havelock, North Carolina"*, we need the bridging evidence like *"Marine Corps Air Station Cherry Point"*. The second challenge is how to reason over multiple pieces of scattered evidence to generate factual-coherent questions.

Previous works mainly focus on single-hop QG, which use neural network based approaches with the sequence-to-sequence (Seq2Seq) framework. Different architectures of encoder and decoder have been designed (Nema et al., 2019; Zhao et al., 2018) to incorporate the information of answer and context to do single-hop reasoning. To the best of our knowledge, none of the previous works address the two challenges we mentioned above for multi-hop QG task. The only work on multi-hop QG (Chauhan et al., 2020) uses multi-task learning with an auxiliary loss for sentence-level supporting fact prediction, requiring supporting fact sentences in different paragraphs being labeled in the training data. While labeling those supporting facts requires heavy human labor and is time-consuming, their method cannot be applied to general multi-hop QG cases without supporting facts.

In this paper, we propose a novel architecture named Multi-Hop Encoding Fusion Network for Question Generation (MulQG) to address the aforementioned challenges for multi-hop QG. First of all, it extends the Seq2Seq QG framework from sing-hop to multi-hop for context encoding. Additionally, it leverages a Graph Convolutional Network (GCN) on an answer-aware dynamic entity graph, which is constructed from entity mentions in answer and input paragraphs, to aggregate the potential evidence related to the questions. Moreover, we use different attention mechanisms to imitate the reasoning procedures of human beings in multi-hop generation process, the details are explained in Section 2.

We conduct the experiments on the multi-hop QA dataset HotpotQA (Yang et al., 2018) with our model and the baselines. The proposed model outperforms the baselines with a significant improvement on automatic evaluation results, such as BLEU (Papineni et al., 2002). The human evaluation results further validate that our proposed model is more likely to generate multi-hop questions with high quality in terms of *Fluency*, *Answerability* and *Completeness* scores.

Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to tackle the challenge of multi-hop reasoning over paragraphs without any sentence-level information in QG tasks.
- We propose a new and effective framework for Multi-hop QG, to do context encoding in multiple hops(steps) with Graph Convolutional Network (GCN).
- We show the effectiveness of our method on both automatic evaluation and human evaluation, and we make the first step to evaluate the model performance in multi-hop aspect.

2 Methodology

The intuition is drawn from human's multi-hop question generation process (Davey and McBride, 1986). Firstly, given the answer and context, we skim to establish a general understanding of the texts. Then, we find the mentions of entities in or correlated to the answer from the context, and analyse nearby sentences to extract useful evidence. Besides, we may also search for linked information in other paragraphs to gain a further understanding of the entities. Finally, we coherently fuse our knowledge learned from the previous steps and start to generate questions.

To mimic this process, we develop our **MulQG framework**. The encoding stage is achieved by a novel **Multi-hop Encoder**. At the decoding stage, we use maxout pointer decoder as proposed in Zhao et al. (2018). The overview of the framework is shown in Figure 1.

2.1 Multi-hop Encoder

Our Multi-hop Encoder includes three modules: (1) Answer-aware context encoder (2) GCN-based entity-aware answer encoder (3) Gated encoder reasoning layer.

The context and answer are split into word-level tokens and denoted as $c = \{c_1, c_2, ..., c_n\}$ and $a = \{a_1, a_2, ..., a_m\}$, respectively. Each word is represented by the pre-trained GloVe embedding (Pennington et al., 2014). Furthermore, for the words in context, we also append the answer tagging embeddings as described in Zhao et al. (2018). The context and answer embeddings are fed into two bidirectional LSTM-RNNs separately to obtain their initial contextual representations $C_0 \in R^{d \times n}$ and $A_0 \in R^{d \times m}$, in which d is the hidden state dimension in LSTM.

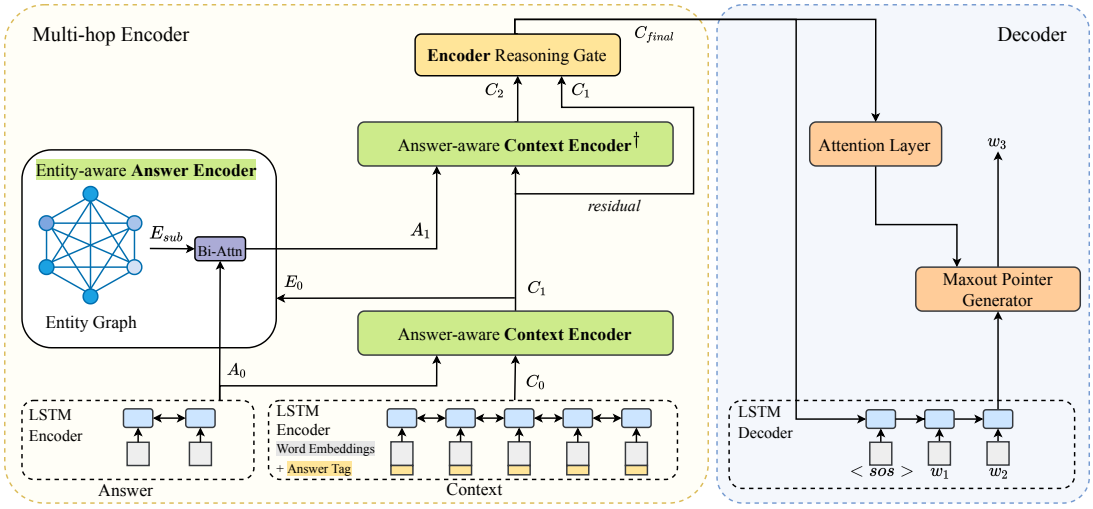

Figure 1: Overview of our MulQG framework. In the encoding stage, we pass the initial context encoding C_0 and answer encoding A_0 to the *Answer-aware Context Encoder* to obtain the first context encoding C_1, then C_1 and A_0 will be used to update a multi-hop answer encoding A_1 via the *GCN-based Entity-aware Answer Encoder*, and we use A_1 and C_1 back to the *Answer-aware Context Encoder*† to obtain C_2. The final context encoding C_{final} are obtained from the *Encoder Reasoning Gate* which operates over C_1 and C_2, and will be used in the max-out based decoding stage.

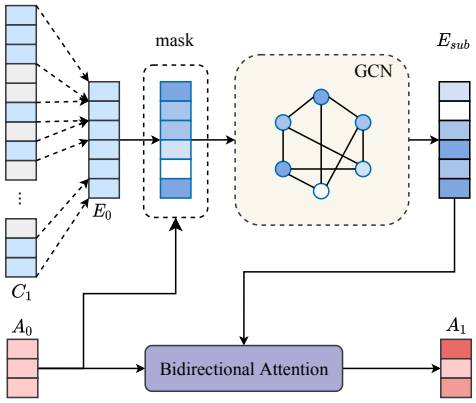

Figure 2: The illustration of GCN-based Entity-aware Answer Encoder.

2.1.1 Answer-aware Context Encoder

Inspired by the co-attention reasoning mechanism in previous machine reading comprehension works (Xiong et al., 2016), we compute the answer-aware context representation via the following steps:

$$S = C_0^T A_0 \in R^{n \times m} \quad (1)$$

$$S' = \text{softmax}(S) \in R^{n \times m} \quad (2)$$

$$S'' = \text{softmax}(S^T) \in R^{m \times n} \quad (3)$$

$$A_0' = C_0 \cdot S' \in R^{d \times m} \quad (4)$$

$$\tilde{C}_1 = [A_0; A_0'] \cdot S'' \in R^{2d \times n} \quad (5)$$

$$C_1 = \text{BiLSTM}([\tilde{C}_1; C_0]) \in R^{d \times n} \quad (6)$$

Firstly, we compute an alignment matrix S (Eq.1), and normalize it column-wise and row-wise to get two attention matrices S' (Eq.2) and S'' (Eq.3). S' represents the relevance of each answer token over the context, and S'' represents the relevance of each context token over the answer. The new answer representation A_0' w.r.t. the context is obtained by Eq.4. Next, the answer dependent context representation is calculated by concatenating old and new answer representations and times the attention weight matrix S'' (Eq.5). Finally, to deeply incorporate the interaction between answer and context, we feed the answer dependent representation $\tilde{C}_1$ combined with original C_0 into a bi-directional LSTM and obtain the answer-aware context encoding C_1 (Eq.6).

2.1.2 GCN-based Entity-aware Answer Encoder

As shown in Figure 2, in order to obtain the multi-hop answer representation, we first compute the entity encoding from the answer-aware context encoding C_1, then we apply GCN to propagate multi-hop information on the answer-aware sub-graph. Finally we obtain the updated answer encoding A_1 via bi-attention mechanism.

Entity Graph Construction The entity graph is constructed with the name entities in context

as nodes, where we use BERT-based name entity recognition model to recognize name entities from the context. The edges are created for the entity pairs if they are in the same sentence, or appear in the same paragraphs. We also connect the entities from each paragraph title to entities within the same paragraph.

Entity Encoding With the answer-aware context encoding C_1 obtained from Answer-aware Context Encoder, we use a mapping matrix M to calculate the entity encoding. M is a binary matrix where $M_{i,j} = 1$ if the i-th token in the context is within the span of the j-th entity. Each entity's encoding will be calculated via a mean-max pooling applied over it's corresponding context token encoding span. $E_0 = \{e_1, e_2, ..., e_g\} \in R^{2d \times g}$, where g is the number of entities, and $2d$ is the dimension since we directly concatenate the mean-pooling and max-pooling encoding.

Answer-aware GCN First we calculate an answer-aware sub-graph, where irrelevant entities are masked out, only those entity nodes related to answer are allowed to disseminate information. Similar to Xiao et al. (2019), a soft mask $M = [m_1, m_2, ..., m_g]$ is calculated via Eq. 7, where each m_i indicate the relatedness of the entity i to the answer, and then apply M on the original graph entities to obtain answer-aware dynamic sub entities graph E_{sub} via Eq. 8.

$$M = \sigma(a_0^T \cdot V \cdot E_0) \in R^{1 \times g} \quad (7)$$

$$E_{sub} = M \cdot E_0 \quad (8)$$

where V is a linear projection matrix and a_0 is the mean pooling over answer encoding A_0, and σ is sigmoid function.

Then we calculate the answer-aware sub-graph's attention matrix as described in Veličković et al. (2017) $A_G = \{\alpha_{i,j}\} \in R^{g \times g}$, where $\alpha_{i,j}$ represents the information that will be assigned from entity i to it's neighbor j, and obtain the one-layer information propagation over the sub-graph via:

$$E_1 = \text{ReLU}(A_G \cdot E_{sub}) \quad (9)$$

The computation from Eq. 9 can be repeated for multiple times to obtain multi-hop entity representation E_M.

Multi-hop Answer Encoding we use bi-attention mechanism (Seo et al., 2016) regarding entities on the sub-graph as memories to update our multi-hop answer encoding A_1 via:

$$A_1 = \text{BiAttention}(A_0, E_M) \quad (10)$$

2.1.3 Encoder Reasoning Gate

We apply a gated feature fusion module on the answer-aware context representations C_1 and C_2 from previous context encoder hops, to keep and forget information to form the final context representation C_{final} via:

$$C_{final} = g_t \odot C_1 + (1 - g_t) \odot C_2 \quad (11)$$

$$g_t = \sigma(w_2^T C_2 + w_1^T C_1 + w_0^T C_0 + b) \quad (12)$$

2.2 Maxout Pointer Decoder

Uni-directional LSTM model is utilized as the decoder of our model. Moreover, we introduce the Maxout Pointer proposed by Zhao et al. (2018) into the decoder for sake of reducing the repetitions in the generation. Pointer Generator enables the decoder to generate the next output token by either computing from the generative probabilistic distribution over the vocabulary or copying from the input sequence. To compute the copy score, the attention over the input sequence which has a vocabulary of V from the current decoder hidden state is leveraged. For the Maxout Pointer Generator, instead of leveraging all the attention score over the input tokens, only the maximal is taken into consideration to avoid the repetitions caused by the input tokens (as it's shown in Eq. 13, where $a_{t,k}$ annotates the decoder-encoder attention score).

$$sc^{copy} = \begin{cases} \max\limits_{k, \text{where } x_k = y_t} a_{t,k} & , y_t \in V \\ -inf & , \text{otherwise} \end{cases} \quad (13)$$

2.3 Breadth-First Search Loss

In addition to the cross-entropy loss, we also introduce Breadth-First Search (BFS) Loss (Xiao et al., 2019) which is a weakly supervised loss to further assist the training procedure. Given the answer entities, we conduct the BFS over the adjacent matrices of the entity graph we build to obtain heuristic masks as a weak supervision signal. The BFS loss is calculated via binary cross-entropy loss between the predicted soft masks M in GCN-based Entity-aware Answer Encoder (Section 2.1.2) and the heuristic masks using Eq. 14 to encourage the model to learn the answer-aware dynamic entity graph better.

$$Loss = L_{CrossEntropy} + \lambda L_{BFS} \quad (14)$$

Model	n-gram						QBLEU4	Answer-ability
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR		
RefNet[1] **(Nema et al., 2019)**	29.79	19.58	14.41	11.10	30.94	18.59	51.80	70.40
MP-GSN∗ **(Zhao et al., 2018)**	34.38	23.00	17.05	13.18	31.85	19.67	48.10	64.60
MulQG	40.08	26.58	19.61	15.11	35.35	20.24	53.90	72.70
MulQG + BFS loss	**40.15**	**26.71**	**19.73**	**15.20**	**35.30**	**20.51**	**54.00**	**72.80**

Table 2: Performance comparison between our MultQG model and state-of-the-art QG models on HotpotQA test set. [1]The results are obtained with the original implementation of RefNet model. We also follows all the hyper-parameter settings as they are described in the paper.

Setting	n-gram						QBLEU4	Answer-ability
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR		
MulQG (our model)	40.08	26.58	19.61	15.11	35.35	20.24	53.90	72.70
MulQG (1-layer GCN)	37.55	25.44	18.95	14.70	34.21	20.56	53.60	72.10
w/o GEAEnc	36.62	24.80	18.50	14.36	33.53	20.39	52.10	70.50
w/o GEAEnc + ACEnc	37.85	26.19	20.15	16.21	33.35	17.86	53.40	71.90
w/o ERG	36.33	24.47	18.14	14.01	33.44	20.28	53.20	71.70
w/o GEAEnc + ACEnc + ERG	34.01	22.95	17.09	13.26	31.90	19.90	52.40	70.70

Table 3: Ablation Study of QG performances on HotpotQA test set, with different encoder modules removed. (Here **GEAEnc**: ***G**raph-based **E**ntity-aware **A**nswer **Enc**oder*, **ACEnc**: ***A**nswer-aware **C**ontext **Enc**oder*[†], **ERG**: ***E**ncoder **R**easoning **G**ate)*

where λ here is a heuristic number and can be selected using cross-validation.

3 Experiment

3.1 Dataset

To demonstrate the performance of our model, we conduct the experiments using HotpotQA (Yang et al., 2018) dataset in an opposite manner. In the QG task, paragraphs and the answers are considered as input, while the corresponding questions are the expected output. HotpotQA is a multi-hop question answering dataset, which contains Wikipedia-based question-answer pairs, with each question requiring multi-hop reasoning across multiple paragraphs to infer the answer. There are mainly two types of multi-hop reasoning in the HotpotQA dataset: *bridge* and *comparison*. Focusing on the multi-hop ability of our model, we filter out all the *yes/no* data samples in the dataset and run our experiments using the remaining corresponding train and test set, which consists of 73k questions in the training set and 8k in the test set.

3.2 Baselines

Since multi-hop QG has been under explored so far, there are very few existing baselines for our comparison. We choose the following two models because of their high relevance with our task and relatively superior performance:

MP-GSN is the first QG model to demonstrate a large improvement with paragraph-level inputs for single-hop QG proposed by Zhao et al. (2018). While they conducted their experiments on SQuAD (Rajpurkar et al., 2016), we use exactly the same experiment settings provided in their configuration file on HotpotQA dataset.

RefNet is the first work that has reported results on HotpotQA dataset for QG proposed by Nema et al. (2019). However, their inputs based on the gold supporting sentences, which contains the facts related to the multi-hop question, and no paragraph-level results have been shown. We experiment with the code they released on paragraphs-level, and test their model's performance on both their validation set and test set of HotpotQA dataset.

We also fine-tuned large pre-trained language models UniLM (Dong et al., 2019) and BART (Lewis et al., 2019) on the multi-hop QG task as comparison benchmark, to further show the effectiveness of our method. The details and the results will be covered in Appendix.

Model	Fluency	Answerability	Completeness	Multi-hop
Baseline	2.26 (0.50)	2.08 (0.87)	2.30 (0.79)	51.5%
Ours	2.46 (0.43)	2.49 (0.61)	2.83 (0.33)	72.3%
Human	2.57 (0.43)	2.67 (0.41)	2.86 (0.26)	81.2%

Table 4: The results of Human Evaluation. The mean values and the standard deviations of the first three evaluation scores, along with the percentage of questions assessed as multi-hop type are shown above.

3.3 Implementation Details

Our word embeddings are initialized by glove.840B.300d (Pennington et al., 2014) and we keep our vocab size as 45000. We use two-layer bi-directional LSTMs for encoder and two-layer uni-directional LSTMs for decoder, and the hidden size is 300 for all the models. We use stochastic gradient descent (SGD) as the optimizer. The initial learning rate is 0.1, and it is reduced during the training stage using a cosine annealing scheduler (Loshchilov and Hutter, 2016). The batch size is 12 and the beam size is 10. We set the dropout probability for LSTM to 0.2 and 0.3 for GCN. The maximum number of epochs is set to 20. We set the maximum number of entities in each context to 80, and we use a two-layer GCN in our GCN-based answer encoder module. After training the model for 10 epochs, we further fine-tune the MulQG model with the help of BFS loss, where the λ in Eq.14 is set to 0.5.

3.4 Automatic Evaluation

3.4.1 Metrics

We use the metrics in previous work on single-hop QG to evaluate the generation performance of our model, with n-gram similarity metrics BLEU[1] (Papineni et al., 2002), ROUGE-L (LIN, 2004), and METEOR using the package released in Lavie and Denkowski (2009). We also quantify the QBLEU4 (Nema and Khapra, 2018a) and answerability score of our models, which was shown to correlate significantly better with human judgements (Nema and Khapra, 2018b).

3.4.2 Results and Analysis

Table 2 shows the performance of various models on the HotpotQA test set. We report the both results of the experiments on our proposed model before and after fine-tuning with auxiliary BFS loss. As it's shown in the table, our MulQG model perform much better than the two baselines methods, with regard to all those measuring metrics, which indicates that the multi-hop procedure can significantly boost the quality of the encoding representations and thus improve the multi-hop question generation performance. Also the BFS loss can further improve the system performance by encouraging learning the answer-aware dynamic entity graph better, which is a key and bottleneck module in the MulQG model.

3.4.3 Ablation Study

To further evaluate and investigate the performance of different components in our model, we perform the ablation study. As we can see from Table 3, both the *GCN-based entity-aware answer encoder* module and *Gated Context Reasoning* module are important to the model. Each of them provides a relative contribution of 2%-3% for overall performance improvement.

w/o GEAEnc: Without *GCN-based Entity-aware Answer Encoder*, answer-related multi-hop evidence information cannot be identified. Without multi-hop answer encoding being updated, next step's answer-aware context encoding will be affected and thus the performance will drop a lot.

w/o GEAEnc + ACEnc: The performance continues to decrease but not that much. This matches with our expectation, since without an informative input A_1 containing multi-hop information from the *GCN-based Entity-aware Answer Encoder*, the *Answer-aware Context Encoder*† cannot generate an informative C_2. Thus remove it won't hurt the performance that much.

w/o ERG: When we remove the *Encoder Reasoning Gate*, the performance drops by around 3% in BLEU-1. This also matches our intuition since without effective feature reasoning and fusion, all the previous encoders cannot generate effective representations. Thus the generation performance will be affected.

w/o GEAEnc + ACEnc + ERG: Without the three modules, the performance directly drops to

[1] https://github.com/Maluuba/nlg-eval

0	-0.0018	House of Many Ways
0	0.0007	Diana Wynne Jones
0	-0.0004	Howl's Moving Castle
0	-0.0017	Castle in the Air
0	-0.0010	Howl's Moving Castle
0	0.0012	British
0	0.0009	Diana Wynne Jones
1	0.0007	Academy Award for Best Animated Feature

1	-0.0028	Prudence Jane Goward
0	0.3688	Australian
0	-0.4160	Mental Health
0	-0.0602	Medical Research
0	0.0361	New South Wales Legislative Assembly
0	0.0170	Goulburn
0	-0.2563	Liberal Party of Australia
0	0.1605	2007

0	0.0048	Jeremy Lee Renner
0	0.1639	American
0	-0.0356	Academy Award for Best Supporting Actor
0	0.4070	2016
0	0.1450	American
0	0.5070	Denis Villeneuve
0	0.4875	Amy Adams
1	1.0000	Jeremy Renner

Example I

Paragraph A:	House of Many Ways is a young adult fantasy novel written by Diana Wynne Jones. The story is set in the same world as "Howl's Moving Castle" and "Castle in the Air".
Paragraph B:	Howl's Moving Castle is a fantasy novel by British author Diana Wynne Jones. ... In 2004 it was adapted as an animated film of the same name, which was nominated for the academy award for best-animated feature.
Answer:	academy award for best animated feature
Baseline:	house of many ways is a young adult fantasy novel written by diana wynne jones , the story is set in the same world as " howl 's moving castle
Ours:	what award was the author of the book house of many ways nominated for ?
Human:	house of many ways is a young adult fantasy novel set in the same world as a novel that was adapted as an animated film of the same name and nominated for what ?

Example II

Paragraph A:	Prudence Jane Goward (born 2 September 1952 in Adelaide), an Australian politician, ... she has previously served as the minister for mental health, minister for medical research, and assistant minister for health between April 2015 and January 2017. ... Goward is a member of the new south wales legislative assembly representing Goulburn for the liberal party of Australia since 2007.
Paragraph B:	Goulburn is an electoral district of the legislative assembly in the Australian state of new south wales. It is represented by Pru Goward of the liberal party of Australia.
Answer:	jane goward
Baseline:	goulburn is an electoral district of the legislative assembly in the australian state of new south wales , it is represented by pru goward of the liberal party of australia
Ours:	which member of the electoral district of goulburn has previously served as the minister for mental health?
Human:	which australian politician represented electoral district of goulburn

Example III

Paragraph A:	Jeremy Lee Renner (born January 7, 1971) is an American actor. ... He was nominated for the academy award for best supporting actor for his much-praised performance in "The Town".
Paragraph B:	Arrival is a 2016 American science fiction film directed by Denis Villeneuve ... It stars Amy Adams, Jeremy Renner, and Forest Whitaker, ...
Answer:	jeremy renner
Baseline:	which american actor starred in the 2016 american science fiction film directed by denis villeneuve ?
Ours:	which star of the movie arrival was nominated for the academy award for best supporting actor for his performance in " the town "?
Human:	name the actor who has acted in the film arrival and who has been nominated for the academy award for best supporting actor for the film " the town " ?

Figure 3: Case study of three examples from the HotpotQA test set. The left part of the figure shows the importance of the entitie nodes, where he left column in red indicates the answer entities and the right colume in blue displays the importance of the entities of graph reasoning at the starting point by the shade of color. The tables show the generated questions from different models along with the corresponding paragraphs and the answer. Moreover, we highlight the reasoning paths of our proposed model in green for a more intuitive display. We also use wavy lines to mark out the snippets of the paragraphs that the questions generated by the MP-GSN model derive from.

single-hop QG system level, which proves the contributions of the whole proposed model.

MulQG (1-layer GCN): When apply 1-layer GCN and only allow information propagation being limited to each node's neighbor, the answer-related evidences might not be able to be fully obtained, thus the performance are not as good as our 2-layer GCN-based model.

3.5 Human Evaluation

Human evaluation is conducted to further analyze the performance of our model (Table 4). We compare the generated questions from MP-GSN model, our model and gold ones on four metrics: ***Fluency***, ***Completeness***, ***Answerability*** and whether the generated questions are ***multi-hop question*** or not. Fluency emphasizes the grammar correctness of the question, while Completeness only focuses on the sentence completeness. Answerability mainly indicates the relationship between the answers and the generated questions. For the first three index, the score for each data sample could be chosen from {1,2,3} in comparison with the other samples generated from the other two models with the same input, where a higher score indicates a better performance on that matrix, For the multi-hop evaluation, we only carry out binary discrimination. We randomly sample 100 data samples from the test set. Ten annotators are asked in total to evaluate them on the aforementioned four metrics. Each sample is evaluated by three different annotators.

To present a more convincing analysis, we conduct the t-test on the human evaluation results. All the reported results between our proposed model and the baseline are statistically significant with a p-

value$<$0.05. We also calculate the inter-annotator agreement using Fleiss' Kappa (Fleiss, 1971) measure and achieve high agreement scores on the proposed model. We observe that our MulQG model largely outperforms the MP-GSN model in terms of Fluency, Answerability and Completeness with more stable quality. Moreover, our model tends to generate more complete question and achieve comparable completeness score with the human annotations. For the multi-hop evaluation, we outperform the strongest baseline by 20.8% on the multi-hop evaluation.

3.6 Case Study

We present a case study comparing between the strong baseline MP-GSN model, our model and the human annotations. Three cases are presented in Figure 3. In the first two examples, it's clearly shown in the examples that the baseline model tends to copy a contiguous and long span of context as the generation, while our proposed model performs better in this aspect. we observe that since the supporting fact information is not leveraged in our method, the generated questions from our model may show a different reasoning path with that for the gold question. There could be multiple ways to construct a multi-hop question given the same input. So the generations may be much different from the gold label, although they are still correct questions, which could be indicated from the first two examples. This phenomenon causes a lower score in automatic matrices, such as BLEU and METEOR, but we note that the generated questions still follow the multi-hop scheme and can be answered with the given answers.

In *Example III*, we show the data sample in an easier mode. In this case, while the answer entity is in one paragraph, a similar entity (annotated with orange color) also appears in another paragraph, which gives a strong clue of the reasoning path and makes it easier for the model to attend to both paragraphs. The generations from our model and the human annotation show almost the same reasoning path. However, we observe that the question generated by MP-GSN model still tends to attend to the entities that are closer to the answer entities. Moreover, for the human annotation in *Example I* and *Example III*, the gold questions have a problem with fluency, which is harmful for the QG models, but interestingly, even with training using these labels, our model is still capable of generating relatively fluent outputs.

4 Related Work

Question Generation Early single-hop QG use rule-based methods to transform sentences to questions (Labutov et al., 2015; Lindberg et al., 2013). Recently neural network based approaches adopt the sequence-to-sequence (Seq2Seq) based framework, with different types of encoders and decoders have been designed (Zhou et al., 2017; Nema et al., 2019; Zhao et al., 2018). Zhao et al. (2018) proposes to incorporate paragraph level content by using Gated Self Attention and Maxout pointer networks, while Nema et al. (2019) proposes a model which contains two decoders where the second decoder refines the question generated by the first decoder using reinforcement learning. There are different ways to attend answer information to the context encoding stage. Zhou et al. (2017) and Liu et al. (2019) directly concatenate answer tagging with the context embedding, while Nema et al. (2019) also applies bi-attention mechanism proposed by Seo et al. (2016) for QA to do answer-aware context representation. Chen et al. (2019) is the most recent work which proposes a reinforcement learning based graph-to-sequence (Graph2Seq) model which use a bidirectional graph encoder on a syntax-based graph for QG, while they still focus on the single-hop QG.

Multi-hop QA Popular Graph Nueral Network (GNN) frameworks, such as graph convolutional networks (Kipf and Welling, 2016), graph attention network (Veličković et al., 2017), and graph recurrent network (Song et al., 2018) have been explored and showed promising results on multi-hop QA task that requiring reasoning. Xiao et al. (2019) proposes a dynamic fused graph network to work on multi-hop QA on the HotpotQA dataset. De Cao et al. (2018) proposes an entity-GCN method to reason over across multiple documents for multi-hop QA on the WIKIHOP dataset (Welbl et al., 2018).

5 Conclusion

Multi-hop QG task is more challenging and worthy of exploration compared to conventional single-hop QG. To address the additional challenges in multi-hop QG, we propose MulQG, which does *multi-hop context encoding* with Graph Convolutional Network and *encoding fusion* via a Gated Reasoning module. To the best of our knowledge,

we are the first to tackle the challenge of multi-hop reasoning over paragraphs without any sentence-level information. The model performance on HotpotQA dataset demonstrates its effectiveness on aggregating scattered pieces of evidence across the paragraphs and fusing information effectively to generate multi-hop questions. The strong reasoning ability of the Multi-hop Encoder in the MulQA model can potentially be leveraged in complex generation tasks for the future work.

References

Hardik Chauhan, Asif Ekbal, Pushpak Bhattacharyya, et al. 2020. Reinforced multi-task approach for multi-hop question generation. *arXiv preprint arXiv:2004.02143*.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Reinforcement learning based graph-to-sequence model for natural question generation. *arXiv preprint arXiv:1908.04942*.

Beth Davey and Susan McBride. 1986. Effects of question-generation training on reading comprehension. *Journal of Educational Psychology*, 78(4):256.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2018. Question answering by reasoning across documents with graph convolutional networks. *arXiv preprint arXiv:1808.09920*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13063–13075.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

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, Los Angeles, California. Association for Computational Linguistics.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898.

Alon Lavie and Michael J Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23(2-3):105–115.

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. *arXiv preprint arXiv:1910.13461*.

C-Y LIN. 2004. Rouge: A package for automatic evaluation of summaries. In *Proc. of Workshop on Text Summarization Branches Out, Post Conference Workshop of ACL 2004*.

David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114.

Bang Liu, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. 2019. Learning to generate questions by learningwhat not to generate. In *The World Wide Web Conference*, pages 1106–1118. ACM.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Preksha Nema and Mitesh M Khapra. 2018a. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959.

Preksha Nema and Mitesh M. Khapra. 2018b. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

Preksha Nema, Akash Kumar Mohankumar, Mitesh M Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2019. Let's ask again: Refine network for automatic question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3305–3314.

Boyuan Pan, Hao Li, Ziyu Yao, Deng Cai, and Huan Sun. 2019. Reinforced dynamic reasoning for conversational question generation. *arXiv preprint arXiv:1907.12667*.

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*, pages 311–318. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Thomas Scialom, Benjamin Piwowarski, and Jacopo Staiano. 2019. Self-attention architectures for answer-agnostic neural question generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6027–6032.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Heung-Yeung Shum, Xiao-dong He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for amr-to-text generation. *arXiv preprint arXiv:1805.02473*.

Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Yunxuan Xiao, Yanru Qu, Lin Qiu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. *arXiv preprint arXiv:1905.06933*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Kaichun Yao, Libo Zhang, Tiejian Luo, Lili Tao, and YanJun Wu. 2018. Teaching machines to ask questions. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4546–4552. AAAI Press.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.

A Appendix

A.1 Detailed Experiment Settings

We run our experiments on 1 GeForce® GTX 1080 Ti GPU, with batch size to 12. The average runtime for our model is around 7500s for one epoch. The total numbers of parameters for our model is : 84250510, while we freeze the word embedding parameters, so our total number of parameters need to be optimized is 57250510. We run the baselines also on the same computing environment, using the configuration file they provided. For the Maxout Pointer baseline, we use a batch size of 16 to fit with our GPU memory.

A.2 Comparison with fine-tuning large pre-trained language models

In order to further show the effectiveness of our method, we further fine-tuned UniLM (Dong et al., 2019) and BART (Lewis et al., 2019) on the multihop QG task. UniLM and BART has obtained state-of-the-art performance on the summarization tasks and also on question generation task on SQuAD (Rajpurkar et al., 2016) dataset.

As we can see from Table A1, the performance of our model is on par with fine-tuning the large-pretrained models on the multihop QG tasks. While our model is much more light-weight and can provide explicit reasoning interpretability.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
Finetune-UniLM(l48_p0_b1)	42.37	29.95	22.61	17.61	40.34	25.48
Finetune-BART(test.hypo.l32_p0_b5)	41.41	30.90	24.39	19.75	36.13	25.20
MulQG	40.08	26.58	19.61	15.11	35.35	20.24
MulQG + BFS loss	40.15	26.71	19.73	15.20	35.30	20.51

Table A1: Performance comparison between our MultQG model and fine-tuning state-of-the-art large pre-trained models on HotpotQA test set.