SGCM: Salience-Guided Context Modeling for Question Generation

Chuyao Ding, Yu Hong*, Jianmin Yao
School of Computer Science and Technology, Soochow University, SuZhou, China
{dddddcyy, tianxianer}@gmail.com, jyao@suda.edu.cn

Abstract

We tackle Paragraph-level Question Generation (abbr., PQG) in this paper. PQG is a task of automatically generating questions given paragraphs and answers. Identifying the relevant sentences to answers is crucial for reasoning the possible questions before generation. Accordingly, we propose a salience-guided approach to enhance PQG. Specifically, we construct an auxiliary task of identifying salient sentences that manifest relevance. Grounded on this auxiliary task and the main task of PQG, we strengthen the BART encoder during training within a multi-task learning framework. In particular, we utilize the identified salient sentences as an explicit guidance to enable the salience-aware attention computation in the BART decoder. We experiment on the benchmark dataset FairytaleQA. The test results show that our approach yields substantial improvements compared to the BART baseline, achieving the $Rouge\text{-}L$, $BLEU4$, $BERTScore$, $Q\text{-}BLUE\text{-}3$ and $F1$-scores of about 56.56%, 19.78%, 61.19%, 54.33% and 43.55%, respectively. Both the source codes and models will be publicly available.

Keywords: Question Generation, Salience Guidance, Comprehension Types

1. Introduction

PQG aims to generate a question for an answer conditioned on the given paragraph (Rus et al., 2010). PQG is different from the Factoid-based Question Generation (FQG) tasks (Song et al., 2018; Nema et al., 2019; Li et al., 2019a; Jia et al., 2020; Wang et al., 2022b; Wu et al., 2022). It is characterized as two aspects as follows:

- The available context for PQG is a paragraph which, on the one hand, contains richer hints for reasoning the question, on the other hand, possessing more noisy information. Most of the FQG tasks deal with sentences (Song et al., 2018; Li et al., 2019a; Jia et al., 2020).
- The answers in PQG are empirically written by annotators, instead of being extracted from the paragraphs. Consequently, the answers may not occur in the paragraphs (Su et al., 2022; Zhao et al., 2022; Wang et al., 2023).

Therefore, a PQG model is required to have the reasoning ability besides of anti-noise encoding capacity. Accordingly, the current studies of PQG tend to investigate the deep reasoning approaches with the aim to generate complex or even multi-hop questions (Pan et al., 2020; Cheng et al., 2021; Fei et al., 2022), where graph-based models (e.g., GAT and Att-GGNN) are used to extract the reasoning chains for question decoding. These approaches have achieved significant improvements.

Nevertheless, graph-based reasoning heavily relies on the qualified chains that consist of relevant nodes (known as token or entity-level clues) as well as exact relations. As a result, the errors caused

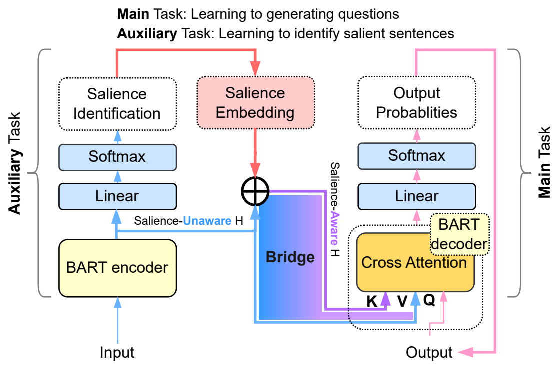


Figure 1: SGCM framework and learning strategy

by the reasoning chain extractor (Stanovsky et al., 2018), graph builder (Qiu et al., 2019; Shi and Lin, 2019; Fan et al., 2019) or entity recognizer (Manning et al., 2014) mislead the reasoning process. This negatively influences the generators.

In this paper, we propose SGCM, a PQG model which reasons questions using salient sentences instead of graph-based evidence chains. Within a multi-task learning framework, SGCM is taught to not only identify salient sentences but generate questions in terms of their salient information. During generation, the salience-aware reasoning is implemented by simply bridging BART encoder and decoder (Lewis et al., 2020) using salience-aware embeddings, where the in-between cross attention is computed. Briefly, we use the encoded salient sentences as explicit evidence to guide the question generation process.

We experiment on FairytaleQA. The test results show that our approach yields substantial improvements compared to the baseline, and outperforms the previous work at all the evaluation metrics.

* Corresponding Author.

LREC-COLING 2024, pages 14755–14762
20-25 May, 2024.

2. Approach

We follow Wang et al. (2022a) to construct SGCM. The framework is shown in Figure 1, where BART is used. BART encoder serves to encode the input answer, question type and paragraph, producing a **salience-unaware** representation. Conditioned on this representation, BART decoder autoregressively generates the tokens of a possible question.

In SGCM, we use multi-task learning to enhance BART, where PQG is the main task, while Salient Sentence Identification (SSI) is the auxiliary task. An additional linear layer with Softmax is connected to BART encoder for SSI. Conditioned on the salient and non-salient sentences determined by SSI, we incorporate the embeddings of the labels "*salient*" and "*non-salient*" into the salience-unaware representation, producing a **salience-aware** representation. On this basis, we deliver both salience-aware and salience-unaware representations to BART decoder, which are adopted as key (**K**) and value (**V**) to guide the cross attention calculation in BART decoder (see the "**Bridge**" in Figure 1).

During training, all the neural networks in SGCM (BART and linear layers) are optimized with the objectives of both PQG and SSI.

2.1. Training Data for SSI

The PQG corpora such as FairytaleQA (Xu et al., 2022) barely provide the annotated salient or non-salient sentences. As a result, a SSI model cannot be trained. To address the issue, we use a heuristic method to produce pseudo-annotated data.

Given a paragraph $\mathcal{G}$ in the training set and a pair of ground-truth Question and Answer (QA pair) in $\mathcal{G}$, we divide all the sentences in $\mathcal{G}$ into two classes (salient and non-salient classes) according to the relevance between the QA pair and each sentence. Unlike a sentence, the QA pair is heterogeneous, possessing a natural interrogative sentence and a fine-grained answer (i.e., token, phrase, entity or short text span). To calculate relevance between homogeneous data, we convert the QA pair into a declarative sentence. We use Demszky et al. (2018)'s QA2D toolkit[1] for conversion. We manually verified the quality of 500 sentences that are produced by QA2D. Fluency is considered as the gold standard in evaluating the quality. The proportion of satisfying instances is 95.6%.

We estimate relevance by $Rough$-L based F1-score (Lin, 2004)[2], a metric of measuring sequence similarity. By this metric, two sequences obtain a higher score if they share a larger longest common subsequence. Accordingly, we determine a sentence as the salient case only if it has a higher sequence similarity with the converted QA pair than a threshold η ($\eta \simeq$0.51). Otherwise, they are determined as the non-salient case.

In this way, we deal with all sentences in a paragraph, and thus obtain two classes of pseudo-annotated data (salient or non-salient sentences).

2.2. Training BART Encoder with SSI

We refer BART encoder to BART$_{\rm en}$ for short, which comprises 6 transformer encoder blocks (Vaswani et al., 2017). The input of BART$_{\rm en}$ is constructed by concatenating a paragraph $\mathcal{G}$, target answer $\mathcal{A}$ and the designated question type $\mathcal{T}$. It is noteworthy that FairtaleQA is used for type-specific PQG. We use BART$_{\rm en}$ to compute the hidden states of $\mathcal{G}$, $\mathcal{A}$ and $\mathcal{T}$: [$\mathcal{H}^{\mathcal{G}}$,$\mathcal{H}^{\mathcal{A}}$,$\mathcal{H}^{\mathcal{T}}$]= BART$_{\rm en}$($\mathcal{G}$, $\mathcal{A}$, $\mathcal{T}$).

A noteworthy detail is that the paragraph is reconstructed before it is input as $\mathcal{G}$. During reconstruction, each sentence in the paragraph is prefixed with a [MASK] token M_i. By BART$_{\rm en}$, the hidden state $\mathcal{H}_i^{\mathcal{M}} \in \mathbb{R}^{1\times d}$ of each M_i is computed. We regard $\mathcal{H}_i^{\mathcal{M}}$ as the hidden state of the i-th sentence. Accordingly, $\mathcal{H}^{\mathcal{G}}$ comprises the hidden state of every token in $\mathcal{G}$ as well as that of each sentence.

We take the $\mathcal{H}_i^{\mathcal{M}}$ of each sentence from $\mathcal{H}^{\mathcal{G}}$. We feed it into a linear layer $\mathcal{D}$ with Softmax to predict the probabilities $\check{y}_i^{\mathcal{M}}$ of being salient or non-salient:

$$\check{y}_i^{\mathcal{M}} = \mathrm{Softmax}\left(\mathcal{D}(\mathcal{H}_i^{\mathcal{M}}, \theta)\right) \quad (1)$$

where, θ denotes the trainable parameters in the linear layer. We use the cross-entropy loss function during optimizing BART$_{\rm en}$ and the linear layer:

$$\mathcal{L}^{\mathcal{M}} = -\frac{1}{N}\frac{1}{\dot{N}}\sum_{i=1}^{N}\sum_{j=1}^{\dot{N}} y_{ij}^{\mathcal{M}} log\left(\check{y}_i^{\mathcal{M}}\right) \quad (2)$$

where, N is the batch size, while $\dot{N}$ is the number of sentences in a paragraph. $y_{ij}^{\mathcal{M}}$ is a one-hot probability vector. It indicates whether the j-th sentence in the i-th paragraph is a salient case. The pseudo-annotated data (Section 2.1) is used to affirm $y_{ij}^{\mathcal{M}}$. By equations (1) and (2), we obtain a binary classifier of SSI, which labels a sentence with the tag "salient" or "non-salient".

2.3. Salience-Aware Hidden States

We produce a salience-aware representation for the input [$\mathcal{G}$, $\mathcal{A}$, $\mathcal{T}$]. It is implemented by incorporating the embeddings of the salient and non-salient tags into the hidden states [$\mathcal{H}^{\mathcal{G}}$,$\mathcal{H}^{\mathcal{A}}$,$\mathcal{H}^{\mathcal{T}}$] output by BART$_{\rm en}$. In this process, $\mathcal{H}^{\mathcal{A}}$ and $\mathcal{H}^{\mathcal{T}}$ are frozen, while $\mathcal{H}^{\mathcal{G}}$ is updated as $\tilde{\mathcal{H}}^{\mathcal{G}}$ by information fusion.

Specifically, assume that a sentence in $\mathcal{G}$ is assigned a salient tag τ by SSI, thus the embedding $h^{\tau} \in \mathbb{R}^{1\times d}$ of τ is fused with the hidden state

[1]https://github.com/kelvinguu/qanli

[2]https://github.com/google-research/google-research/tree/master/rouge

$h^{\mathcal{G}} \in \mathbb{R}^{1\times d}$ ($h^{\mathcal{G}} \in \mathcal{H}^{\mathcal{G}}$) of every token in the sentence. Element-wise aggregation is used for fusion: $h^{\tau} \oplus h^{\mathcal{G}}$. If the sentence is assigned a non-salient tag $\bar{\tau}$, the above information fusion is conducted using the embedding $h^{\bar{\tau}}$ of $\bar{\tau}$. Both the embeddings h^{τ} and $h^{\bar{\tau}}$ are randomly initialized.

To facilitate reading, we refer the original output [$\mathcal{H}^{\mathcal{G}}$,$\mathcal{H}^{\mathcal{A}}$,$\mathcal{H}^{\mathcal{T}}$] of BART$_{\text{en}}$ to $\mathcal{H}$, while the salience-aware version [$\tilde{\mathcal{H}}^{\mathcal{G}}$,$\mathcal{H}^{\mathcal{A}}$,$\mathcal{H}^{\mathcal{T}}$] to $\tilde{\mathcal{H}}$.

2.4. Salience-guided BART Decoder

As shown in the bridge of SGCM in Figure 1, we deliver both the output $\mathcal{H}$ of BART$_{\text{en}}$ and the salience-aware version $\tilde{\mathcal{H}}$ to BART decoder (abbr., BART$_{de}$). They are used to guide the unmasked multi-head self-attention computation in BART$_{\text{de}}$, where $\tilde{\mathcal{H}}$ is used as the Key, while $\mathcal{H}$ the Value.

In practice, we employ a 6-layer BART$_{\text{de}}$ which possesses 6 transformer decoder blocks as well as a linear layer with Softmax. The produced hidden states $\mathcal{H}$ and $\tilde{\mathcal{H}}$ at the last layer of BART$_{\text{en}}$ will be delivered to each layer of BART$_{\text{de}}$. At each time of delivering $\mathcal{H}$ and $\tilde{\mathcal{H}}$, the salience-guided attention computation is executed. On this basis, we use BART$_{\text{de}}$ to autoregressively generate the tokens of the possible question:

$$p(\check{\mathcal{Y}}_i|\check{\mathcal{Y}}_{1:i-1}) = \mathrm{Softmax}\left(\check{\mathcal{D}}\left(\mathrm{BART}_{de}\left(\mathcal{H},\tilde{\mathcal{H}},\check{\mathcal{Y}}_{1:i-1},\vartheta\right)\right)\right) \tag{3}$$

where $\check{\mathcal{Y}}_{1:i-1}$ denotes the token predicted at the earlier i-1 steps, while ϑ is all the learnable parameters in the decoding channel. $\check{\mathcal{D}}$ is the accompanying linear layer of BART$_{\text{de}}$.

During training, teacher-forcing learning (Toomarian and Barhen, 1992) is used for optimization. Accordingly, the cross-entropy based loss of PQG is calculated as follows:

$$\mathcal{L}^{\mathcal{Q}} = -\frac{1}{\ddot{N}}\sum_{i=1}^{\ddot{N}} p(\mathcal{Y}_i|\mathcal{Y}_{1:i-1}) \tag{4}$$

where, $\ddot{N}$ is the number of tokens in the ground-truth question $\mathcal{Q}^G$. $p(\mathcal{Y}_i|\mathcal{Y}_{1:i-1})$ is the probability that the i-th token of $\mathcal{Q}^G$ is generated during decoding. The in-batch loss is calculated as $\sum_N \mathcal{L}^{\mathcal{Q}}$.

We train the networks in both encoding and decoding channels within a multi-task learning framework, where PQG and SSI serve as the primary and auxiliary tasks respectively. The combined loss of the two tasks is calculated as $\mathcal{L} = \mathcal{L}^{\mathcal{M}} + \mathcal{L}^{\mathcal{Q}}$.

3. Experimentation

3.1. Experimental settings

Datasets– We experiment on FairytaleQA (Xu et al., 2022) under two schemes, namely Skill and HCD. In Skill (Wang et al., 2023), there are 5 question types considered for evaluating type-aware PQG, including *Remember*, *Understand*, *Analyze*, *Create* and *Evaluate*. In HCD (Zhao et al., 2022), there are 3 types used for evaluation, including *Action*, *Casual relationship* and *Outcome resolution*. We follow Wang et al. (2023) and Zhao et al. (2022) to split FairytaleQA into the training, validation and test sets without any change.

Evaluation– We evaluate PQG models using BLEU-4 (Papineni et al., 2002), BERTScore (Zhang et al., 2020), Q-BLEU-3 (Nema and Khapra, 2018), Rouge-L (Lin, 2004) and $Rough$-L based $F1$-score (Lin, 2004). They are abbreviated as B4, BES, Q-B3, R-L and F1 when performance is reported.

Hyperparameters– The maximum length of input is set to 520. The size in beam search is set to 8. We use a learning rate of 6.25e-5 and batch size of 16 for the Skill scheme. For HCD, the learning rate is set to 5e-6 and batch size is set to 1.

Model	R-L	B4	BES	Q-B3
SkillQG* (Wang et al., 2023)	55.23	19.49	59.91	52.96
BART$_{base}$ (baseline)	54.68	18.92	59.99	52.01
BART$_{base}$+SGCM	**56.56**	**19.78**	**61.19**	**54.33**

Table 1: Performance on the Skill test split.

Model	Precision	Recall	F1
ECQG (Zhao et al., 2022)	37.80	31.54	30.58
ECQG+GT-type (Zhao et al., 2022)	46.48	31.96	35.77
BART$_{large}$ (baseline)	48.77	40.92	42.30
BART$_{large}$+SGCM	**50.15**	**42.15**	**43.55**

Table 2: Performance on HCD test split.

3.2. Results and Analysis

In our experiments, we firstly compare with Wang et al. (2023)'s **SkillQG** which obtains the state-of-the-art performance for Skill scheme. SkillQG is characterized as the utilization of entity-related knowledge generated by GPT-2 (Radford et al., 2019). Due to the use of BART$_{base}$ (Lewis et al., 2020) as backbone in SkillQG, we specify BART$_{base}$ as the baseline during comparison. In addition, we compare with Zhao et al. (2022)'s Event-Centric QG (**ECQG**) which obtains a noticeable effect for HCD scheme. In ECQG, two BARTs are used, one of which generates question-type-specific summaries, the other performs PQG grounded on the generated summaries. A prototypical ECQG uses the predicted question types to guide automatic summarization, while its updated version (namely **ECQG+GT-type**) uses ground-truth question types for guidance. Due to the use of BART$_{large}$ (Lewis et al., 2020) as backbone in ECQG, we use BART$_{large}$ as the baseline when HCD scheme is followed.

The PQG performance obtained under Skill and HCD schemes is shown in Tables 1 and 2, respectively. It can be observed that our SGCM yields

Model	B4	Meteor	R-L
SGGDQ-DP (Pan et al., 2020)	15.53	20.15	36.94
DCQG (Cheng et al., 2021)	15.26	19.99	-
CQG (Fei et al., 2022)	25.09	27.45	41.83
QA4QG$_{large}$ (Su et al., 2022)	25.70	27.44	**46.48**
BART$_{base}$+SGCM	**26.16**	**28.51**	44.06

Table 3: Performance on HotpotQA for SFT.

Model	B4	Meteor	R-L
QA4QG$_{base}$ (Su et al., 2022)	19.68	24.55	40.44
QA4QG$_{large}$ (Su et al., 2022)	21.21	25.53	**42.44**
BART$_{base}$+SGCM	**22.61**	**26.04**	40.61

Table 4: Performance on HotpotQA for FDC.

substantial improvements compared to BART$_{base}$ and BART$_{large}$. Besides, SGCM outperforms SkillQG, ECQG and ECQG+GT-type. The advantage of SGCM is attributed to the avoidance of omitting inherent knowledge or absorbing external interference. By contrast, SkillQG suffers from the interference of external noises occurring in the generated knowledge, while ECQG omits a part of paragraph when the summary is merely used.

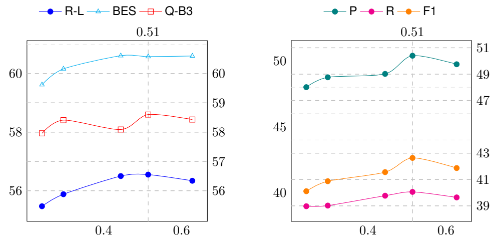
3.3. Generality of SGCM

We additionally verify the generality of SGCM by evaluating it on the other PQG corpus. The multi-hop QA corpus HotpotQA (Yang et al., 2018) is used for verification, where the evaluation schemes of both SFS and FDC are considered. In SFS, a PQG model is allowed to generate questions from **Supporting Fact Sentences**, without being disturbed by irrelevant contexts. In FDC, **Full Document Context** is forcibly used for PQG. The state-of-the-art PQG models on HotpotQA are compared to our SGCM, including **SQQDQ-DP** (Pan et al., 2020), **DCQG** (Cheng et al., 2021), **CQG** (Fei et al., 2022) and **QA4QG** (Su et al., 2022).

The PQG performance for SFT and FDC on HotpotQA is shown in Table 3 and 4. It can be observed that our method (BART$_{base}$+SGCM) outperforms SGGDQ-DP, DCQG and CQG for all the common metrics of B4, Meteor (Lavie and Agarwal, 2007) and R-L. Technically, our SGCM doesn't rely on reasoning chains, while SGGDQ-DP, DCQG and CQG do. This difference reveals the possible reasons that the latter models perform worse, including 1) out-of-chain contexts still contain rewarding evidence for question reasoning, and 2) some unqualified chains misguide the reasoning process.

3.4. Detecting Reliable Thresholds

Constructing the pseudo-annotated dataset (Section 2.1) is crucial for training BART encoder within the auxiliary task SSI. The setting of the threshold η plays the most important role during the pseudo-annotation process. Instead of detecting the proper



(a) Skill (b) HCD

Figure 2: Detecting an effective threshold η.

Model	R-L	B4	BES	Q-B3
BART$_{base}$+SGCM	**56.56**	19.78	**61.19**	54.33
BGE-based	55.97	**20.87**	60.50	**54.70**
BLEURT-based	56.10	20.32	60.72	53.82

Table 5: Performance on the Skill test split with different relevance computation.

η in a separate task, we integrate it with the development process of our PQG model. Specifically, the reliability of η is indirectly determined conditioned on the effects it has on the performance of PQG.

Figure 2 shows the effects it has when validation set is used for metric calculation, where both the schemes of Skill and HCD are considered. It can be found that the best setting of η is at 0.51, where the PQG model reaches the best development performance for most of evaluation metrics (R-L, Q-B3 as well as Rough-L based Precision, Recall and F1-score). Samely, we set the η for HotpotQA as 0.48 for both SFT and FDC settings.

3.5. Computing Relevance

Besides $Rough$-L based $F1$-score (Lin, 2004), we employ BGE(Xiao et al., 2023)[3] and BLEURT(Sellam et al., 2020)[4] to compute relevance on Skill dataset, selecting the optimal thresholds of 0.65 and 0.52, respectively. The remaining experimental settings are consistent with SGCM. The test results are shown in Table 5. It can be observed that models trained on the BGE-based and BLEURT-based data exhibit a slight performance fluctuation on different metrics. Accordingly, it is proved that our salience-guided method is general to any data annotation tool.

4. Related Work

The previous studies concentrate on factoid questions. The answers are contiguous spans occurred in the paragraph-level contexts (Nema et al., 2019; Jia et al., 2020; Wang et al., 2022b). Recently, some QG tasks allow answers to be out-of-context,

[3]https://github.com/FlagOpen/FlagEmbedding
[4]https://github.com/lucadiliello/bleurt-pytorch

which requires deep reasoning of possible questions. In particular, generating multi-hop questions attracts an intense interest, where the generator is required to reason relations among constituents in a complex syntactic structure (Pan et al., 2020; Cheng et al., 2021; Fei et al., 2022; Su et al., 2022).

Cognitive levels are subsequently observed in the above studies. This inspires the exploration of QG for different types of questions that imply diverse cognition of human (Yao et al., 2022; Dugan et al., 2022; Eo et al., 2023). Meanwhile, detecting and summarizing reliable facts for reasoning complex questions has been studied, which plays a crucial role of supplying salient evidence during decoding (Zhao et al., 2022; Wang et al., 2023).

Combining latent information of evidence enables the initial guidance to the decoder of QG. The previous work generally uses a gated attention module (Zhao et al., 2018; Li et al., 2019b; Jia et al., 2021) for information fusion. Different from prior studies, we utilize the salient information as an explicit guidance to enable the salience-aware attention computation in the decoder.

5. Conclusion

We utilize a multi-task learning method to enhance BART based paragraph-level question generation. The auxiliary task of identifying salient sentences is used to highlight reliable evidence for reasoning questions. It facilitates a soft anti-noise reasoning process, without forcibly filtering non-salient sentences. Experiments on FairytaleQA show that our approach yields substantial improvements compared to the BART baseline, and outperforms the previous arts. More importantly, we demonstrate the generality of our model by the verification on the other corpus, i.e., HotpotQA, where multi-hop questions are required to be generated. In the future, we will use this approach to construct salience-aware thought chains, where non-salient chains will be paid less attention instead of being filtered.

6. Acknowledgements

We thank all anonymous reviewers for their insightful comments. This work is supported by National Science Foundation of China (62376182, 62076174).

7. References

Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin, and Yefeng Zheng. 2021. Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5968–5978. Association for Computational Linguistics.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *CoRR*, abs/1809.02922.

Liam Dugan, Eleni Miltsakaki, Shriyash Upadhyay, Etan Ginsberg, Hannah Gonzalez, Dayheon Choi, Chuning Yuan, and Chris Callison-Burch. 2022. A feasibility study of answer-agnostic question generation for education. *CoRR*, abs/2203.08685.

Sugyeong Eo, Hyeonseok Moon, Jinsung Kim, Yuna Hur, Jeongwook Kim, Songeun Lee, Changwoo Chun, Sungsoo Park, and Heuiseok Lim. 2023. Towards diverse and effective question-answer pair generation from children storybooks. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6100–6115. Association for Computational Linguistics.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 4184–4194. Association for Computational Linguistics.

Zichu Fei, Qi Zhang, Tao Gui, Di Liang, Sirui Wang, Wei Wu, and Xuanjing 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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6896–6906. Association for Computational Linguistics.

Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2020. How to ask good questions? try to leverage paraphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6130–6140. Association for Computational Linguistics.

Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2021. EQG-RACE: examination-type question generation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13143–13151. AAAI Press.

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, WMT@ACL 2007, Prague, Czech Republic, June 23, 2007*, pages 228–231. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Jingjing Li, Yifan Gao, Lidong Bing, Irwin King, and Michael R. Lyu. 2019a. Improving question generation with to the point context. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3214–3224. Association for Computational Linguistics.

Jingjing Li, Yifan Gao, Lidong Bing, Irwin King, and Michael R. Lyu. 2019b. Improving question generation with to the point context. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3214–3224. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60. The Association for Computer Linguistics.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3950–3959. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3312–3321. Association for Computational Linguistics.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1463–1475. Association for Computational Linguistics.

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 of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6140–6150. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai C. Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge. In *INLG 2010 - Proceedings of the Sixth International Natural Language Generation Conference, July 7-9, 2010, Trim, Co. Meath, Ireland*. The Association for Computer Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7881–7892. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018. Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 569–574. Association for Computational Linguistics.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 885–895. Association for Computational Linguistics.

Dan Su, Peng Xu, and Pascale Fung. 2022. QA4QG: using question answering to constrain multi-hop question generation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pages 8232–8236. IEEE.

Nikzad Benny Toomarian and Jacob Barhen. 1992. Learning a trajectory using adjoint functions and teacher forcing. *Neural Networks*, 5(3):473–484.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Fei Wang, Kaiqiang Song, Hongming Zhang, Lifeng Jin, Sangwoo Cho, Wenlin Yao, Xiaoyang Wang, Muhao Chen, and Dong Yu. 2022a. Salience allocation as guidance for abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 6094–6106. Association for Computational Linguistics.

Qifan Wang, Li Yang, Xiaojun Quan, Fuli Feng, Dongfang Liu, Zenglin Xu, Sinong Wang, and Hao Ma. 2022b. Learning to generate question by asking question: A primal-dual approach with uncommon word generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 46–61. Association for Computational Linguistics.

Xiaoqiang Wang, Bang Liu, Siliang Tang, and Lingfei Wu. 2023. Skillqg: Learning to generate question for reading comprehension assessment. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13833–13850. Association for Computational Linguistics.

Zichen Wu, Xin Jia, Fanyi Qu, and Yunfang Wu. 2022. Enhancing pre-trained models with text structure knowledge for question generation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 6564–6574. International Committee on Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *CoRR*, abs/2309.07597.

Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Nora Bradford, Branda Sun, Tran Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022. Fantastic questions and where to find them: Fairytaleqa - an authentic dataset for narrative comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 447–460. Association for Computational Linguistics.

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Bingsheng Yao, Dakuo Wang, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Mo Yu, and Ying Xu. 2022. It is ai's turn to ask humans a question: Question-answer pair generation for children's story books. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 731–744. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 3901–3910. Association for Computational Linguistics.

Zhenjie Zhao, Yufang Hou, Dakuo Wang, Mo Yu, Chengzhong Liu, and Xiaojuan Ma. 2022. Educational question generation of children storybooks via question type distribution learning and event-centric summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5073–5085. Association for Computational Linguistics.