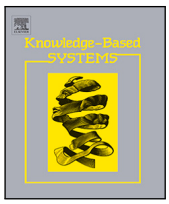
Knowledge-Based Systems 300 (2024) 112120

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



RTRL: Relation-aware Transformer with Reinforcement Learning for Deep Question Generation

Hongwei Zeng [a,*], Bifan Wei [b,d], Jun Liu [b,c]

[a] BYD Auto Engineering Research Institute, Shenzhen, Guandong, 518118, China
[b] School of Computer Sciences and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China
[c] National Engineering Lab for Big Data Analytics, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China
[d] School of Continuing Education, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China

ARTICLE INFO

Keywords:
Deep question generation
Relation-aware transformer
Reinforcement learning

ABSTRACT

Deep question generation is an important yet challenging NLP task where a multi-step reasoning chain connecting multiple documents is required to answer the questions. In this paper, we propose a Relation-aware Transformer with Reinforcement Learning (RTRL) to improve the performance of deep question generation. Specifically, RTRL first utilizes the relation-aware Transformer to capture both the explicit linguistic and implicit semantic relations for better text representation. Besides, RTRL utilizes the sentence-level reward in a reinforcement learning framework to deal with the metric inconsistency between training and evaluation, and thus improve evaluation metrics with reference questions. Experimental results on the HotpotQA dataset show that RTRL outperforms baselines, highlighting its effectiveness in generating deep questions.

1. Introduction

The Question Generation (QG) task takes documents and answers as inputs to generate natural language questions. The questions it generates can be used to test learners' knowledge mastery, which plays a very important role in the field of education. In other fields such as dialogue systems [1,2], information retrieval [3,4], and iterative learning control [5,6], question generation systems can also effectively increase the size of training data and further enhance model performance for corresponding tasks. Existing QG works can be categorized into the Shallow Question Generation (SQG) [7,8] and Deep Question Generation (DQG) [9,10]. Different from SQG, which assumes the question can be answered with simple reasoning or text matching, DQG requires combining disjoint pieces to support the reasoning chain of the deep question. Although many well-designed neural models [11,12] have achieved advanced performance on the inverse of multi-hop question answering dataset [13], these methods still face many challenges. Fig. 1 illustrates a motivating example. We need the information from both documents *ElecTRONica* and *Disney California Adventure* to infer the deep question with the given answer. There are mainly two issues that should be considered to capture relevant contents scattered in multiple documents: (i). How to identify answer-related contents in documents, which is important to alleviate random generation; (ii). How to aggregate contents from multiple documents to support the generation of deep questions.

To tackle these issues, first, we need to extract better contextual representation by capturing the answer information. As we can see *Document 1* in Fig. 1, the words related to the question are most likely to be close to the answer, *ElecTRONica*. Specifically, the bridge entity, *Disney California Adventure*, that links the two documents is also near the answer. These suggest that the location information of the answer helps to better understand the document and identify the bridge entity. However, other question generation models do not use answer information in this way. For example, Pan et al. [9] only use the answer information on the decoder to initialize the decoding state. Second, we need to aggregate contents scattered in multiple documents which is essential to build deep questions. As we can see in Fig. 1, the underlined coreference cluster containing mention *Disney California Adventure*, *Disney California Adventure Park*, and *It*, links the two documents. Besides, most of the relevant words (colored) is in the sentences where answer or bridge entity are located and can be parsed through linguistic knowledge, such as dependency parser. These suggest that linguistic relations, such as coreference and dependency, are critical to capturing and aggregating relevant content in multiple documents. However, the previous methods are mostly based on recurrent or convolutional neural networks that are hard to model such explicit relations.

Motivated by the above issues, we propose a Relation-aware Transformer with Reinforcement Learning (RTRL) for deep question generation to address both issues. We first introduce the relative distance to

* Corresponding author.
E-mail addresses: hongwei.zeng@foxmail.com (H. Zeng), weibifan@xjtu.edu.cn (B. Wei), liukeen@xjtu.edu.cn (J. Liu).

https://doi.org/10.1016/j.knosys.2024.112120
Received 30 January 2024; Received in revised form 30 May 2024; Accepted 11 June 2024
Available online 13 June 2024

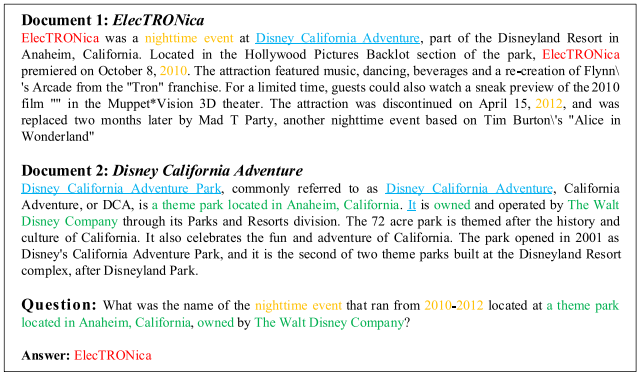


Fig. 1. A multi-hop QA sample from the HotpotQA dataset.

the answer location and embed them into the distribution space. Then we use a bi-directional LSTM encoder to obtain the answer contextualized representation. Similar to network cooperation strategies [14,15], we propose a relation-aware Transformer which takes both explicit linguistic and implicit semantic relations as cooperators by concatenating different attention heads to aggregate important contents from multiple documents. The explicit relations are annotated with pre-trained linguistic parsers while the implicit relations are learned with semantic similarity. Specifically, we incorporate the explicit relation knowledge into the self-attention network with mask strategy to constraint information propagation by utilizing multihead attention strategy for representation fusion. Different attention heads are then fused to form the unified representation. Such model design provides more flexibility for the model to handle diverse structures and information and thus gain more views to achieve a better understanding. Moreover, we propose to utilize reinforcement learning to further optimize and fine-tune the trained model with sentence-level rewards. Such a learning paradigm can alleviate the exposure bias during sequential decoding and metric inconsistency between training and evaluation. Therefore, the model can be directly optimized to fit the desired metrics.

We summarize the main contributions as follows:

- We propose a deep question generation framework which can utilize a relation-aware Transformer with both explicit linguistic and implicit semantic relations.
- We propose to utilize reinforcement learning to optimize the evaluation metric of questions with mixed rewards directly.
- We have done extensive experiments that verify the effectiveness of the proposed RTRL model on the HotpotQA dataset.

The remainder of this paper is organized as follows. In Section 2, we review the related works of question generation, Transformer, and reinforcement learning. Section 3 describes the detailed RTRL framework. Section 4 discusses the experimental settings and results. Finally, the conclusion and future work are presented in Section 5.

2. Relate work

2.1. Question generation

The question generation task which aims to produce natural questions based on given documents and answers, plays a very important role in the NLP field. Early question generation methods were mostly based on manual-crafted rules and templates [7,16,17], which can produce accurate and fluent questions without grammar error. However, these models tend to produce a similar output format which lack of diversity and scalability.

Recently, question generation has benefited from the rapid development of neural networks and deep learning. These methods generally utilize the attention-based encoder–decoder framework [2,8,18] for context encoding and question decoding. Besides, various techniques are employed to improve the quality of generated questions, such as answer information incorporation [19–21], wider contexts modeling [22–24], question-worthy contents selection [25–27]. However, most of these models can only generate shallow questions that can be easily answered by simple text matching.

Inspired by multi-hop question answering [13], research work on Deep Question Generation (DQG) [9,10] has also received more and more attention. DQG requires multi-hop reasoning over complicated input documents to generate more difficult questions. To capture a better understanding of the complicated inputs, [9,11,28,29] proposed to transform natural language sequence into text-attributed graph with linguistic and semantic parser tools. Then they will employ graph neural networks to model these constructed graph structures which can capture the long-range dependencies more efficiently. For example, Pan et al. [9] provided two ways to extract semantic units. One is to extract predicate-parameter tuples as sub-graphs through Semantic Role Labeling (SRL), and the elements of the tuples are used as nodes. Edges will also be introduced if elements of different tuples have an inclusion relationship or refer to the same entity. The second is to reconstruct the dependency parse tree component graph of the sentence, such as removing unimportant symbolic components and merging consecutive nodes that can form a complete semantic unit. Parse trees of different sentences are connected by matching similar nodes. To directly optimize the specified evaluation metrics, [10,30,31] proposed to utilize reinforcement learning with various reward functions. For example, Gupta et al. [30] proposed a question-aware supporting fact prediction task as a reward function, guiding the model to better utilize the information in all supporting facts to generate multi-hop questions through reinforcement learning. To alleviate the low-resource challenge of deep question generation, [32,33] proposed to transfer knowledge from unlabeled or shadow questions. For example, Yu et al. [32] utilized neural hidden semi-Markov models to mine structural patterns on large-scale unlabeled questions, and transfer the mined patterns as prior knowledge to deep question generation to improve model performance.

2.2. Transformer

Transformer [11,34,35] have attracted increasing attention due to their flexibility in parallel computation and long-term dependency modeling. However, no explicit knowledge or positional relations are used in the original version of the Transformer framework [34]. To enhance structural information with prior knowledge, various work [36–39] have been proposed to incorporate explicit relations into self-attention module in Transformer, such as linguistic and commonsense knowledge relation. Some work leverages linguistic knowledge relation, such as dependency and semantic role labeling, to improve the effectiveness of the self-attention mechanism. For example, Strubell et al. [36] presented linguistically-informed self-attention which combines multi-head self-attention with multi-task learning across dependency parsing. Mihaylov et al. [40] dedicated self-attention heads to explicit discourse and semantic annotations as a bias to finding answers to questions in longer contexts. Zhang et al. [38] proposed using syntax to guide the text modeling by incorporating explicit syntactic dependency of interest design into attention mechanism for better linguistically motivated word representations. Bugliarello et al. [41] proposed to incorporate syntactic knowledge into self-attention networks which shows consistent and higher translation quality improvements. To enhance the representation ability of multivariate time series, Xiao et al. [42] proposed a densely knowledge-aware network with a two-branch network representation strategy where residual multihead convolutional network and transformer-based network are utilized to extract local and global patterns respectively. The two-branch views are then fused with self-distillation. The difference between this method and ours is mainly the fusion form of different representation views, where we concatenate different attention masks in the same layer to further increase information interaction.

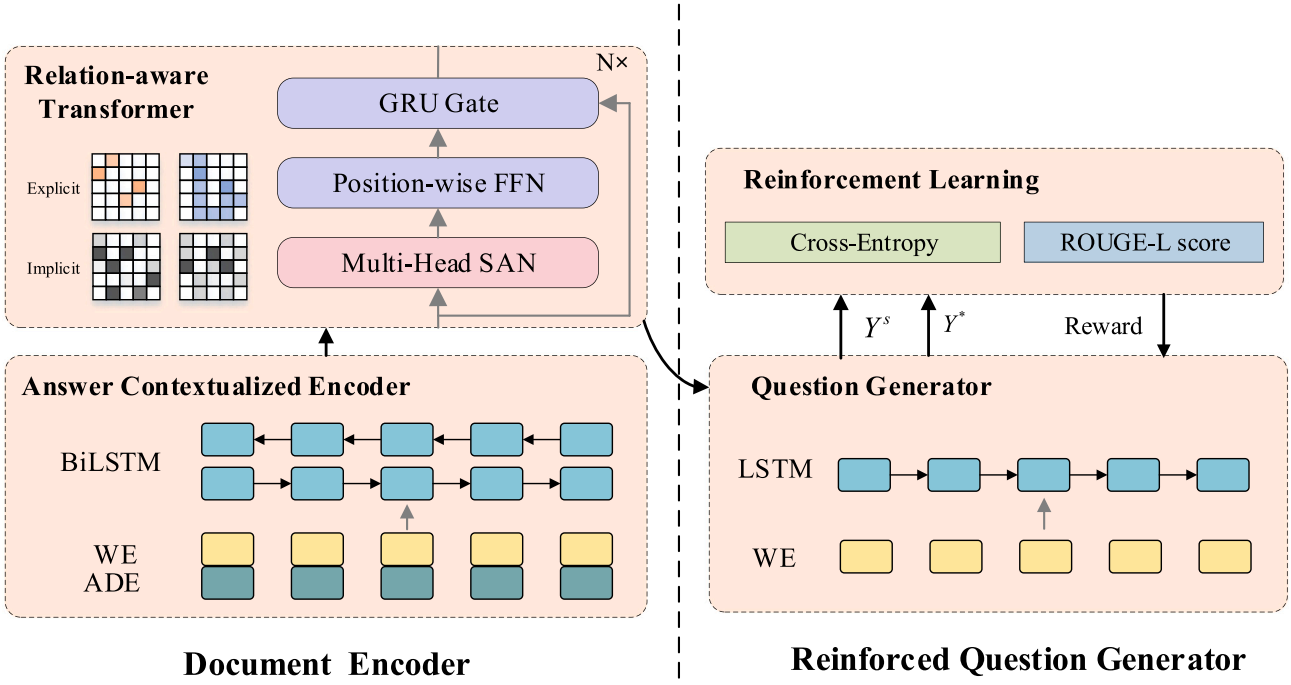


Fig. 2. Overview of our RTRL framework. The input word embedding is concatenated with answer distance embedding. The explicit relations refer to co-reference and constituency respectively, while the implicit relations refer to the full connections between all word pairs. Finally, the trained model will be fine-tuned through reinforcement learning combined with cross-entropy and ROUGE-L score as the reward. Notably, WE, ADE, FFN, and SAN refer to Word Embedding, Answer Distance Embedding, Feed-Forward Network, and Self-Attention Network respectively.

2.3. Reinforcement learning

Reinforcement learning (RL) optimizes discrete decisions making with cumulative rewards, which has been widely utilized across various fields, such as robot path planning [43] recommendation systems [44] computer vision [45] and Natural Language Generation (NLG) [46]. Especially in NLG, RL can solve the inconsistent problem between training and inference and the non-differentiable problem of sentence-level loss. In particular, RL is also utilized to improve the quality of question with question-related rewards. For examples, Yu et al. [47] introduced a reinforcement learning optimized graph-to-sequence model to directly optimize the evaluation metrics, such as BLEU-4, with self-critical sequence training algorithms. To improve the effectiveness and robustness, Wang et al. [10] applied a multi-reward optimization strategy to train the model with the mixture loss of BLEU-4 and word movers distance scores. Xie et al. [31] explored the effectiveness of various question-specific rewards, such as fluency, relevance, and answerability. Besides, they conducted comprehensive analytic experiments in which the rewards are highly correlated with human judgment. Guan et al. [48] utilized semantic-rich rewards, such as naturality and difficulty, which demonstrates the combination of multiple rewards can further improve the performance in evaluation metrics.

3. Methodology

In this section, we provide a detailed description of the proposed Relation-aware Transformer with Reinforcement Learning (RTRL) framework. The overall architecture is illustrated in Fig. 2.

3.1. Answer contextualized encoder

First, all input elements $x = (x_1, \ldots, x_m)$ are embedded in a distributional space as $w = (w_1, \ldots, w_m)$. w_j is a embedding vector of word x_j stored in a lookup table $\mathcal{R}^{|V| \times d_w}$, where $|V|$ is the vocabulary size and d_w is the word embedding size. We represent the input as a matrix $W \in \mathcal{R}^{m \times d_w}$ which is the concatenation of its word embedding. We initialize word embedding with pre-trained GloVe word vectors [49]. The word embeddings for words not in GloVe will be initialized randomly.

Since we observe that the words related to the question are most likely to be close to the answer, we embed the relative distance of each word to the answer to guide the model to capture relevant content. Therefore, we can represent the input x in another distribution space as $p = (p_1, \ldots, p_m)$. $p_j \in \mathcal{R}^{d_w}$ is the answer distance embedding vector which is embedded by the relative distance of word x_j to the answer span. By adding the embedding of each word with corresponding answer distance embedding, we can obtain the answer-aware embedding for the input x as follow:

$$E(x) = [w_1 + p_1, \ldots, w_m + p_m] \in \mathcal{R}^{m \times d_w} \tag{1}$$

Relative position embedding is critical for question generation since they specify our model to ask specific answers about the documents.

To learn answer contextualized word embedding, we utilize a bi-directional LSTM [50] to capture the contextual information from both forward and backward direction by feeding the answer-aware embedding $E(x)$.

$$\{h_1, \ldots, h_m\} = \text{BiLSTM}(\{w_1 + p_1, \ldots, w_m + p_m\}) \tag{2}$$

where h_j are the contextualized word embedding concatenated by the forward and backward hidden states in jth step. Therefore, we can build an answer contextualized embedding for each word based on the entire input text and the target answer.

3.2. Relation-aware transformer

Similar to the origin Transformer network, each relation-aware Transformer block also has two modules. First, the self-attention module allows the model to capture long-term dependencies, assigning higher weights to more relevant parts of the token sequence. The single-head self-attention layer is propagated with the scaled dot-product

attention as following:

$$ATTN_s(Q,K,V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V \tag{3}$$

where Q, K, and V are abbreviation of Query, Key and Value which are calculated with the following linear transformations respectively:

$$\begin{aligned} Q &= H \cdot W^Q \\ K &= H \cdot W^K \\ V &= H \cdot W^V \end{aligned} \tag{4}$$

where $W^Q \in \mathcal{R}^{d\times d_Q}$, $W^K \in \mathcal{R}^{d\times d_K}$, and $W^V \in \mathcal{R}^{d\times d_V}$, d is the dimension of the input vector H. d_Q, d_K, d_V refer to dimension of the output vectors Q, K, and V respectively. H is the hidden states from the previous layer. Specifically, the attention score between hidden states h_i and h_j in k-th head is calculated as following:

$$\alpha^k_{i,j} = \frac{1}{\sqrt{d}}(h_i W^Q_k)(h_j W^K_k)^T \tag{5}$$

The default single-head self-attention network can learn the implicit relation between all word pairs.

To learn the explicit relation with linguistic knowledge, We modified Eq. (5) to calculate the attention score between hidden states h_i and h_j in k'-th head as follows:

$$\alpha^{k'_{i,j}} = \frac{1}{\sqrt{d}}(h_i W^Q_{k'})(h_j W^K_{k'})^T + b^{k'}_{i,j} \tag{6}$$

$$b^{k'}_{i,j} = \begin{cases} 0, & \text{if } (i,j) \in R \\ -\inf, & \text{otherwise} \end{cases} \tag{7}$$

where R refers to co-reference and constituency relations which are annotated with pre-trained linguistic parser and discussed in .

To aggregate the information with both explicit and implicit relations, we apply a multi-head attention network to fuse representations as follows:

$$ATTN_m(Q,K,V) = [\text{Head}_1;\cdots;\text{Head}_H]\cdot \text{W}^H \tag{8}$$

where H is the head number, $\text{W}^H \in \mathcal{R}^{H*d_V\times d}$ is the trainable parameter matrix of a feed-forward neural network, $\text{Head}_i = ATTN_s(Q,K,V)$, d_K and d_V refer to the dimension size of the key and value vector respectively.

The other module is the position-wise Feed-Forward Network (FFN) which contains two linear layers and a non-linear RELU activation layer in between to each position separately and identically.

$$\text{FFN}(x) = \text{ReLU}(x\cdot W_1 + b_1)\cdot W_2 + b_2 \tag{9}$$

where W_1, W_2, b_1, and b_2 are trainable parameters.

Each relation-aware Transformer layer can propagate information from word to word with explicit or implicit relation for one-hop. By stacking the relation-aware transformer multiple layers, each word can attend over information up to multiple hops away. Besides, we adopt the gate mechanism [51] with Gated Recurrent Unit (GRU) [52] to fuse information with different hop mechanism as follow:

$$h^l = \text{GRU}(y^l, x^l) \tag{10}$$

where x^l, y^l are the input and output of l-th Transformer layer respectively. h^l is the fused hidden states in l-th layer and h^0 is the answer contextualized representation.

3.3. Reinforcement learning

The supervised training of question generation task aims to maximize the likelihood of each successive word in the ground-truth questions, which is optimized with the word-level cross-entropy loss over the target vocabulary given the paragraph X and gold history of question words $Y^*_{<t}$ as follows:

$$\mathcal{L}_{SL} = -\sum_t \log P_{\text{final}}(y^*_t) \tag{11}$$

However, this training strategy can cause exposure bias [53] because we use partially generated words instead of golden history words to infer the next word during testing. Besides, the word-level objective function does not match the sequence-level evaluation metrics, so that our model cannot be trained to an optimal solution.

To address the above issues and directly optimize our question generation system with question-related metrics, we transform the generative model into a reinforcement learning task for optimization as in [53]. In this paper, we use the REINFORCE algorithm [54] which enables us to train at the sequence-level using any user-defined and non-differentiable reward functions. Besides, we follow the self-critical sequence training strategy [55] which incorporates a baseline decoded by greedy search to reduce the variance of gradient estimate in the REINFORCE algorithm.

$$\mathcal{L}_{RL} = (r(Y^s, Y^*) - r(Y^b, Y^*))\sum_t \log P_{\text{final}}(y^s_t) \tag{12}$$

where r refers to the reward function which measures the n-gram similarity of the input sentence-pair, Y^s and Y^b are the generated questions by multi-nomial sampling and greedy decoding of the final distribution P_{final} respectively, Y^* is the human annotated question.

We then fine-tune our generative model with a mixed loss function via linearly combination as follows:

$$\mathcal{L}_{\text{mixed}} = \mathcal{L}_{SL} + \lambda\mathcal{L}_{RL} \tag{13}$$

where λ is a hyperparameter that balances the importance of word-level supervised loss $\mathcal{L}_{SL}$ and question-level reinforced loss $\mathcal{L}_{RL}$.

4. Experiments

We evaluate RTRL on HotpotQA [13], a challenging multi-hop question answering dataset. All experiments are conducted in PyTorch framework on a single Tesla V100.

4.1. Experiment setup

Dataset. The HotpotQA dataset [13] contains over 113K Wikipedia-based question–answer pairs where answering each question requires reasoning on multiple supporting facts. Following the similar preprocessing procedures in [9], we concatenate the supporting facts and the sentences which overlap with the ground-truth question as the input context. For efficient training, we remove the examples where the document length is longer than 200 tokens or the question length is longer than 50 tokens. Finally, the processed dataset contains 87,686 training examples and 6,948 testing examples respectively.

Relation Annotations. We mainly consider two linguistic relations including coreference resolution and constituency parsing, which can resolve the reasoning relations between and inside evidence respectively. (i). Coreference resolution aims to cluster all linguistic expressions referring to the same entity, which plays a very important role in multi-hop reasoning. To annotate relations across multiple documents, we utilize the pre-trained co-reference resolution model [56] based on SpanBERT embeddings [57]. The words inside each cluster are fully connected, and there is no connection between other word pairs. (ii). Constituency parsing aims to hierarchically organize a sentence into phrasal constituents with context-free grammar. Similar to [38], we adopt a constituency parser with head-driven phrase structure grammar [58] based on ELMo embeddings [59] to annotate the constituency relation. All the pre-trained models are publicly available in AllenNLP demos.[1]

Hyperparameters. The embedding dimension of the word vector is set to 300. The answer tagging indicators are embedded in 3-dimensional vectors. We use a two-layer stacked BiLSTM with a hidden

[1] https://demo.allennlp.org

size 768 for the encoder, and a single-layer undirected LSTM with a hidden size 768 for the decoder. The relation-aware Transformer consists of 3 stacked layers with the hidden size being 768 and the number of attention heads being 8. The dimensionality of the inner feed-forward layer is 2048. The dropout with probability $p = 0.3$ is applied to the LSTM stacks and the attention module.

Implementation Details. During training, word vectors are initialized with GloVe [49]. The word vectors of words that are out of the vocabulary of GloVe are initialized randomly. The rest model parameters are initialized using Xavier [60]. We utilize the gradient descent algorithm by the Adam [61] optimizer. The learning rate is initialized as 0.00025. We set the mini-batch size as 64 and the maximum training epoch number as 20. During inference, we generate questions with beam search and set the beam width as 10. The decoding stops when generates the end token <EOS> or achieves the maximum decoding steps which is set to 50 in this paper. In order to prevent the generation of too short and meaningless, we set the minimum decoding steps as 6.

Evaluation Metrics. We employ BLEU [62], ROUGE-L [63], and METEOR [64] to measure the quality of generated questions since these metrics are widely used in many natural language generation tasks, including the deep question generation task in this paper. All these metrics are the higher the score, the better. Besides, these metrics correlate to the human judgment of generated questions and thus are appropriate to evaluate the performance of question generation models.

4.2. Baselines

We have compared our proposal with the following representative baseline methods: 6 classical baselines widely used in single-hop question generation (Seq2seq-Attn [8], NQG++ [19], ASs2s [20], S2s-at-mp-gsa [22], CGC-QG [26], IGND [65]), 6 baselines specially designed for deep question generation (SRL-Graph [9], DP-Graph [9], Strong Transformer [11], MulQG [28], ADDQG [10], CQG [12]), and 2 pre-trained language models: (UNiLM [66], BART [67]).

- **Seq2seq-Attn** [8]: Seq2seq-Attn is a widely used baseline method in many natural language generation tasks with sequence input and sequence output. Here, we use LSTM for both input and output sequence modeling. The attention mechanism is used for dynamic context modeling.
- **NQG++** [19]: NQG++ tasks answer position tagging and lexical features as additional word-level feature embedding to enhance the answer and syntactic representations. Besides, NQG++ utilizes copy mechanism to deal with unknown or rare words by copying words from input directly.
- **S2s-at-mp-gsa** [22]: S2s-at-mp-gsa proposed gated self-attention to process long paragraph which enables the model to select relevant content and filter noise more efficiently.
- **ASs2s** [20]: ASs2s replaces the answer in paragraph with special tokens to prevent information leaking and utilized a keyword network for answer encoding to guide the question generation.
- **CGC-QG** [26]: CGC-QG proposed a content selection task to identify which word to copy or generate. The identified results are utilized as an additional feature which is embedded and concatenated with word embedding as input.
- **IGND** [65]: IGND utilized graph neural network to model the rich structure information of previously generated text and copied words iteratively during the decoding.
- **SRL-Graph** [9]: SRL-Graph utilized the semantic role labeling of answer-aware supporting facts to build the semantic graph and employed a gated and attention-based graph neural network to fuse the graph representation.
- **DP-Graph** [9]: DP-Graph is a variant of SRL-Graph where the semantic graph is constructed with dependency relation.
- **Strong Transformer** [11]: Strong Transformer introduced a graph-augmented transformer encoder to leverage relations between entities in the text. We report the version trained with negative log-likelihood.
- **MulQG** [28]: MulQG proposed a graph convolutional network based fusion network to aggregate the scattered evidences for question generation which requires multi-hop reasoning over different documents.
- **ADDQG** [10]: ADDQG proposed a gated connection layer to incorporate answer information for the initialization of the decoder. Besides, both syntactic and semantic rewards are utilized to optimize the evaluation metrics of generated questions directly.
- **CQG** [12]: CQG proposed a controlled generation framework with key entities extractor and flag tag which ensures the complexity of the generated question.
- **UNiLM** [66]: UNiLM is a unified language model for both understanding and generation which has also achieved great performance in question generation tasks.
- **BART** [67]: BART is a denoising autoencoder that is pre-trained by the corrupted document reconstruction with diverse noise transformations.

4.3. Main results

The performance of the baseline and RTRL on the deep question generation is presented in Table 1. Bold and underlined values indicate the optimal and suboptimal scores, respectively. By comparison, we have the following observations:

- RTRL surpasses the baselines regarding all metrics, demonstrating that the proposed method can generate more deep-aware questions by incorporating linguistic knowledge and a reinforcement learning framework. We have conducted a t-test on BLEU-4 and the results show that the improvement is statistically significant (p-value < 0.01). Compared to the best baseline CQG, RTRL achieves absolute improvements of 0.89 and 5.25 points in BLEU-4 and ROUGE-L metrics.
- Compared to the vanilla model Seq2Seq+Attn, the other methods that utilizes answer information, whether it is answer position embedding [19,22,26], separate answer encoding [20], or answer-aware decoder initialization [9,10], perform significantly better. It suggests that incorporating answers indeed improves the quality of generated questions.
- Compared with models designed for shallow QG, models designed for deep QG, such as SRL-Graph, DP-Graph, and ADDQG, all propose to utilize relation based on linguistic knowledge and have achieved much better performance. This indicates that the effective use of linguistic relations between multiple documents has significantly improved the reasoning over the complex question generation task.

4.4. Ablation studies

In this section, we empirically show the effectiveness of the proposed RTRL framework. For this ablation study, we first conduct an ablation experiment on reinforcement learning and then ablate the major components of the overall network architecture. The detailed ablation results are presented in Table 2.

Effectiveness of Reinforcement Learning. We experimentally verify that utilizing reinforcement learning allows for better optimization for deep question generation with sentence-level metrics than traditional word-level cross-entropy objective functions. Compared to the variant without reinforcement learning (-w/o RL), the overall model has an improvement of 0.62 in BLEU-4, 1.05 in ROUGE-L, and 0.11 in METEOR, showing the contribution of reinforcement learning for

Table 1
Automatic evaluation results on the HotpotQA dataset.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
Seq2Seq+Attn	32.97	21.11	15.41	11.81	33.48	18.19
NQG++	35.31	22.12	15.53	11.50	32.01	16.96
ASs2s	34.60	22.77	15.21	11.29	32.88	16.78
S2s-at-mp-gsa	35.36	22.38	15.88	11.85	33.02	17.63
CGC-QG	31.18	22.55	17.69	14.36	40.94	25.20
IGND	41.22	24.71	18.99	16.36	38.34	24.19
UNiLM	42.37	29.95	22.61	17.61	40.34	25.48
BART	41.41	30.90	24.39	19.75	36.13	25.20
SRL-Graph	40.40	26.83	19.66	15.03	36.24	19.73
DP-Graph	40.55	27.21	20.13	15.53	36.94	20.15
Strong Transformer	–	–	–	20.02	39.49	22.40
MulQG	40.15	26.71	19.73	15.20	35.30	20.51
ADDQG	44.34	31.32	22.68	17.54	38.09	20.56
CQG	49.71	37.04	29.93	25.09	41.83	**27.45**
RTRL	**52.12**	**39.04**	**31.40**	**25.98**	**47.08**	26.37

Table 2
Ablation study of the proposed RTRL on the HotpotQA dataset.

Model	BLEU-4	ROUGE-L	METEOR
RTRL	**25.98**	**47.08**	**26.37**
RTRL (-w/o RL)	25.36	45.93	26.26
-*w/o* Answer Contextualized Encoder	14.59	35.53	20.94
-*w/o* Relation-aware Transformer	24.23	45.31	25.63
-*w/o* Relation-aware Mask	24.60	45.74	26.07

alleviating metric inconsistent. It is clearly evidenced that the further incorporation of the RL component produces the highest increase in the ROUGE-L score. Such results are as expected since it fine-tunes the RTRL backbone network and directly optimizes the ROUGE-L score as the reward.

Effectiveness of Answer Contextualized Encoder. This model variant removes both the answer distance embedding and the BiLSTM contextualized encoder, and directly feeds the word embeddings into the transformer network. We can see that the performance of RTRL drops significantly from 25.36 to 14.59 in BLEU-4 compared with the model without Answer Contextualized Encoder. Specifically, this is because the relative distance to the answer can help the model capture the relevant content about what to ask. Besides, the BiLSTM can capture document representations that contain sequence structure and answer information through forward and backward propagation. And because of the sequence propagation characteristics of LSTM, the closer the word is the answer in sequential distance, the greater the influence of the answer, which also helps the model focus on the local context centered on the answer.

Effectiveness of Relation-aware Transformer. This model variant removes the relation-aware transformer module and utilizes the output of BiLSTM as the final encoder features for attentive decoding. We can see the relation-aware Transformer module has achieved 1.13 point improvement in BLEU-4. Compared with the vanilla transformer (-w/o linguistic relation), the performance drops from 25.36 to 24.60 in BLEU-4, which demonstrates the effectiveness of utilizing linguistic relations for multi-hop reasoning. This is because the relation-aware transformer can model both explicit linguistic relations and implicit relations between token pairs. Besides, the stacked relation-aware transformer can attend over information up to multiple hops away. The combined utilization of answer and relation enables the proposed model to generate better deep questions.

Effectiveness of Relation-aware Mask. Given the vanilla Transformer, we explore the effectiveness of the explicit relation mask for deep question generation. Compared to the RTRL (-w/o RL), the performance drops from 25.36, 45.93, and 26.26 to 24.60, 45.74, and 26.07 in BLEU-4, ROUGE-L, and METEOR respectively. The result confirms that the incorporation of explicit linguistic knowledge is important to deep question generation.

Table 3
Sensitive analysis of the number of Transformer blocks.

Layer Number	BLEU-4	ROUGE-L	METEOR
1	24.47	45.52	25.87
2	24.83	45.63	25.54
3	**25.36**	**45.93**	**26.26**
4	24.51	45.72	25.47

Table 4
Human evaluation results.

Model	Fluency	Relevance	Answerability
NQG++	3.12	2.96	2.85
DP-Graph	3.58	3.72	3.34
CQG	4.13	3.83	3.57
RTRL	**4.35**	**4.14**	**4.07**

4.5. Sensitive analysis of transformer layers

To evaluate the sensitivity of Transformer layers, we explore several variants with layer numbers varying from 1 to 4. As shown in Table 3, we can see the BLEU-4 score increases up to a threshold as the Transformer layer number increases from 1 to 3. Besides, all these metrics are better than the baseline which removes the relation-aware Transformer (shown in Table 2). This confirms the effectiveness of the proposed relation-aware Transformer block for deep question generation in a certain range. When the number of layers further increased to 4, we observed a decrease in performance. This may be overfitting caused by an increase in parameters.

4.6. Human evaluation

To evaluate the generated questions in more human-aligned metrics, a human evaluation study has been conducted by scoring models with human annotators. We randomly select 100 samples from the test dataset. Given the input context and answer, we present the deep questions generated by the baselines and RTRL; and then employ three educated volunteers to score the generated questions in three metrics: fluency, relevance, and answerability, respectively. Scores range from 1 to 5, with higher being better. The description of three manual metrics is as follows: (1) fluency measures whether the questions follow the right grammar; (2) relevance measures whether the questions are relevant to the contexts; and (3) answerability measures whether the questions are answerable according to the given contexts.

Table 4 presents the human evaluation results. The key observations can be summarized as follows:

- RTRL significantly outperforms all baselines in terms of fluency, relevance, and answerability, indicating that RTRL is capable

of producing more reasonable and high-quality questions. On the one hand, this performance improvement is because our model architecture uses answer distance embedding and linguistic knowledge to enhance the interaction between answer information and context. On the other hand, reinforcement learning can directly optimize evaluation metrics.

- By comparing Tables 1 and 4, we can see that the performance rankings of different models are the same in both automatic and manual metrics. This conclusion shows that there is a certain positive correlation between the two. The reason is that ground-truth questions score very high in manual metrics, and thus the model can also achieve higher manual metrics scores while the model is optimized to approach ground-truth questions.
- Overall, these model have the highest fluency score, followed by the relevance score, and the lowest answerability score, which illustrates the difficulty of generating answerable questions. Our model RTRL improves the performance by 5.33%, 8.09%, and 14.01% compared to the best baseline CQG in fluency, relevance, and answerability respectively. This difference in performance improvement is reflected in the fact that the baseline model has already achieved good performance in fluency and relevance on the one hand, and on the other hand reinforcement learning can directly optimize the objective function through the sentence-level rewards.

5. Conclusion and future work

In this study, we have proposed a relation-aware Transformer with reinforcement learning for deep question generation. Specifically, we employ an answer contextualized encoder with relative answer distance embedding to enhance answer-related representation and relation-aware Transformer to unify both explicit linguistic relations and implicit semantic relations to enhance the understanding of multiple document contexts. We further apply the reinforcement learning framework with sentence-level rewards to improve the evaluation metrics of generated questions. We have conducted extensive experiments on the HotpotQA dataset which illustrates the superiority of the proposed RTRL in deep question generation in terms of both automatic and manual evaluation.

Currently, the explicit linguistic relations are annotated with pre-trained parsers which means the whole training from scratch is not a complete end-to-end fashion. In future work, we aim to explore graph structure learning to discover the latent structures automatically and investigate more effective ways to model the rich structure information.

CRediT authorship contribution statement

Hongwei Zeng: Software, Methodology, Conceptualization. **Bifan Wei:** Writing – review & editing, Formal analysis. **Jun Liu:** Supervision, Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by National Key Research and Development Program of China (2022YFC3303600), National Natural Science Foundation of China (62137002, 62293553, 62293554, 61937001, and 62176209), Natural Science Basic Research Program of Shaanxi (2023-JC-YB-593).

References

[1] S. Sharma, L. El Asri, H. Schulz, J. Zumer, Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation, CoRR abs/1706.09799, 2017.
[2] T. Shao, F. Cai, W. Chen, H. Chen, Self-supervised clarification question generation for ambiguous multi-turn conversation, Inform. Sci. 587 (2022) 626–641.
[3] H. Zamani, S.T. Dumais, N. Craswell, P.N. Bennett, G. Lueck, Generating clarifying questions for information retrieval, in: Y. Huang, I. King, T. Liu, M. van Steen (Eds.), WWW '20: The Web Conference 2020, ACM / IW3C2, 2020, pp. 418–428.
[4] M. Gaur, K. Gunaratna, V. Srinivasan, H. Jin, ISEEQ: information seeking question generation using dynamic meta-information retrieval and knowledge graphs, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI, 2022, pp. 10672–10680.
[5] S. Guan, Z. Zhuang, H. Tao, Y. Chen, V. Stojanovic, W. Paszke, Feedback-aided PD-type iterative learning control for time-varying systems with non-uniform trial lengths, Trans. Inst. Meas. Control 45 (11) (2023) 2015–2026.
[6] Z. Zhuang, H. Tao, Y. Chen, V. Stojanovic, W. Paszke, An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints, IEEE Trans. Syst. Man Cybern. Syst. 53 (6) (2023) 3461–3473.
[7] M. Heilman, N.A. Smith, Good question! Statistical ranking for question generation, in: Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, The Association for Computational Linguistics, 2010, pp. 609–617.
[8] X. Du, J. Shao, C. Cardie, 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), 2017, pp. 1342–1352.
[9] L. Pan, Y. Xie, Y. Feng, T.-S. Chua, M.-Y. Kan, Semantic graphs for generating deep questions, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 1463–1475.
[10] L. Wang, Z. Xu, Z. Lin, H. Zheng, Y. Shen, Answer-driven deep question generation based on reinforcement learning, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 5159–5170.
[11] D.S. Sachan, L. Wu, M. Sachan, W. Hamilton, Stronger transformers for neural multi-hop question generation, CoRR abs/2010.11374, 2020.
[12] Z. Fei, Q. Zhang, T. Gui, Di Liang, S. Wang, W. Wu, X. Huang, 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, Association for Computational Linguistics, 2022, pp. 6896–6906.
[13] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, C.D. Manning, HotpotQA: A dataset for diverse, explainable multi-hop question answering, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 2369–2380.
[14] X. Song, N. Wu, S. Song, Y. Zhang, V. Stojanovic, Bipartite synchronization for cooperative-competitive neural networks with reaction-diffusion terms via dual event-triggered mechanism, Neurocomputing 550 (2023) 126498.
[15] Z. Zhang, X. Song, X. Sun, V. Stojanovic, Hybrid-driven-based fuzzy secure filtering for nonlinear parabolic partial differential equation systems with cyber attacks, Internat. J. Adapt. Control Signal Process. 37 (2) (2023) 380–398.
[16] M. Heilman, Automatic Factual Question Generation from Text, vol. 195, Language Technologies Institute School of Computer Science Carnegie Mellon University, 2011.
[17] S.F. Kusuma, D.O. Siahaan, C. Fatichah, Automatic question generation with various difficulty levels based on knowledge ontology using a query template, Knowl.-Based Syst. 249 (2022) 108906.
[18] X. Sun, J. Liu, Y. Lyu, W. He, Y. Ma, S. Wang, Answer-focused and position-aware neural question generation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2018, pp. 3930–3939.
[19] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, M. Zhou, Neural question generation from text: A preliminary study, in: National CCF Conference on Natural Language Processing and Chinese Computing, 2017, pp. 662–671.
[20] Y. Kim, H. Lee, J. Shin, K. Jung, Improving neural question generation using answer separation, in: AAAI, 2019, pp. 6602–6609.
[21] X. Ma, Q. Zhu, Y. Zhou, X. Li, Improving question generation with sentence-level semantic matching and answer position inferring, in: AAAI, 2020, pp. 8464–8471.
[22] Y. Zhao, X. Ni, Y. Ding, Q. Ke, 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, 2018, pp. 3901–3910.
[23] L.A. Tuan, D.J. Shah, R. Barzilay, Capturing greater context for question generation, in: Proceedings of the Conference on Artificial Intelligence, 2020, pp. 9065–9072.
[24] H. Zeng, Z. Zhi, J. Liu, B. Wei, Improving paragraph-level question generation with extended answer network and uncertainty-aware beam search, Inform. Sci. 571 (2021) 50–64.

[25] X. Du, C. Cardie, Identifying where to focus in reading comprehension for neural question generation, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2067–2073.
[26] B. Liu, M. Zhao, D. Niu, K. Lai, Y. He, H. Wei, Y. Xu, Learning to generate questions by LearningWhat not to generate, in: The World Wide Web Conference, 2019, pp. 1106–1118.
[27] B. Liu, H. Wei, Di Niu, H. Chen, Y. He, Asking questions the human way: Scalable question-answer generation from text corpus, in: WWW, 2020, pp. 2032–2043.
[28] D. Su, Y. Xu, W. Dai, Z. Ji, T. Yu, P. Fung, Multi-hop question generation with graph convolutional network, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020, 2020, pp. 4636–4647.
[29] X. Ma, Q. Zhu, Y. Zhou, X. Li, D. Wu, Asking complex questions with multi-hop answer-focused reasoning, CoRR abs/2009.07402, 2020.
[30] D. Gupta, H. Chauhan, R.T. Akella, A. Ekbal, P. Bhattacharyya, Reinforced multi-task approach for multi-hop question generation, in: Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, International Committee on Computational Linguistics, 2020, pp. 2760–2775.
[31] Y. Xie, L. Pan, D. Wang, M. Kan, Y. Feng, Exploring question-specific rewards for generating deep questions, in: Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, International Committee on Computational Linguistics, 2020, pp. 2534–2546.
[32] J. Yu, W. Liu, S. Qiu, Q. Su, K. Wang, X. Quan, J. Yin, Low-resource generation of multi-hop reasoning questions, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6729–6739.
[33] X. Huang, J. Qi, Y. Sun, R. Zhang, Latent reasoning for low-resource question generation, in: Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021, in: Findings of ACL, Association for Computational Linguistics, 2021, pp. 3008–3022.
[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[35] Z. Xiao, H. Tong, R. Qu, H. Xing, S. Luo, Z. Zhu, F. Song, L. Feng, CapMatch: Semi-supervised contrastive transformer capsule with feature-based knowledge distillation for human activity recognition, IEEE Trans. Neural Netw. Learn. Syst. (2023).
[36] E. Strubell, P. Verga, D. Andor, D. Weiss, A. McCallum, Linguistically-informed self-attention for semantic role labeling, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 5027–5038.
[37] Q. Guo, X. Qiu, P. Liu, X. Xue, Z. Zhang, Multi-scale self-attention for text classification, in: AAAI, 2020, pp. 7847–7854.
[38] Z. Zhang, Y. Wu, J. Zhou, S. Duan, H. Zhao, R. Wang, SG-net: Syntax-guided machine reading comprehension, in: AAAI, 2020, pp. 9636–9643.
[39] J. Yang, A. Ke, Y. Yu, B. Cai, Scene sketch semantic segmentation with hierarchical transformer, Knowl.-Based Syst. 280 (2023) 110962.
[40] T. Mihaylov, A. Frank, Discourse-aware semantic self-attention for narrative reading comprehension, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2019, pp. 2541–2552.
[41] E. Bugliarello, N. Okazaki, Enhancing machine translation with dependency-aware self-attention, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020, pp. 1618–1627.
[42] Z. Xiao, H. Xing, R. Qu, L. Feng, S. Luo, P. Dai, B. Zhao, Y. Dai, Densely knowledge-aware network for multivariate time series classification, IEEE Trans. Syst. Man Cybern. Syst. 54 (4) (2024) 2192–2204.
[43] Q. Zhou, Y. Lian, J. Wu, M. Zhu, H. Wang, J. Cao, An optimized Q-learning algorithm for mobile robot local path planning, Knowl.-Based Syst. 286 (2024) 111400.
[44] F. Liu, R. Tang, X. Li, W. Zhang, Y. Ye, H. Chen, H. Guo, Y. Zhang, X. He, State representation modeling for deep reinforcement learning based recommendation, Knowl.-Based Syst. 205 (2020) 106170.
[45] N. Le, V.S. Rathour, K. Yamazaki, K. Luu, M. Savvides, Deep reinforcement learning in computer vision: a comprehensive survey, Artif. Intell. Rev. 55 (4) (2022) 2733–2819.
[46] V. Srinivasan, S. Santhanam, S. Shaikh, Using reinforcement learning with external rewards for open-domain natural language generation, J. Intell. Inf. Syst. 56 (1) (2021) 189–206.
[47] Y. Chen, L. Wu, M.J. Zaki, Reinforcement learning based graph-to-sequence model for natural question generation, in: Proceedings of International Conference on Learning Representations, 2020.
[48] M. Guan, S.K. Mondal, H. Dai, H. Bao, Reinforcement learning-driven deep question generation with rich semantics, Inf. Process. Manag. 60 (2) (2023) 103232.
[49] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543.
[50] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[51] E. Parisotto, H.F. Song, J.W. Rae, R. Pascanu, Ç. Gülçehre, S.M. Jayakumar, M. Jaderberg, R.L. Kaufman, A. Clark, S. Noury, M. Botvinick, N. Heess, R. Hadsell, Stabilizing transformers for reinforcement learning, in: Proceedings of the International Conference on Machine Learning, Vol. 119, 2020, pp. 7487–7498.
[52] J. Chung, Ç. Gülçehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, CoRR abs/1412.3555, 2014.
[53] M. Ranzato, S. Chopra, M. Auli, W. Zaremba, Sequence level training with recurrent neural networks, in: International Conference on Learning Representations, 2016.
[54] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. 8 (1992) 229–256.
[55] S.J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, V. Goel, Self-critical sequence training for image captioning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1179–1195.
[56] K. Lee, L. He, M. Lewis, L. Zettlemoyer, End-to-end neural coreference resolution, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2017, pp. 188–197.
[57] M. Joshi, D. Chen, Y. Liu, D.S. Weld, L. Zettlemoyer, O. Levy, SpanBERT: Improving pre-training by representing and predicting spans, Trans. Assoc. Comput. Linguist. 8 (2020) 64–77.
[58] V. Joshi, M.E. Peters, M. Hopkins, Extending a parser to distant domains using a few dozen partially annotated examples, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2018, pp. 1190–1199.
[59] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 2227–2237.
[60] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.
[61] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, 2015.
[62] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: A Method for Automatic Evaluation of Machine Translation, Association for Computational Linguistics, 2002, pp. 311–318.
[63] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Association for Computational Linguistics Workshop, 2004, pp. 74–81.
[64] S. Banerjee, A. Lavie, METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005, pp. 65–72.
[65] Z. Fei, Q. Zhang, Y. Zhou, Iterative GNN-based decoder for question generation, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, Association for Computational Linguistics, 2021, pp. 2573–2582.
[66] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, H. Hon, Unified language model pre-training for natural language understanding and generation, in: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 13042–13054.
[67] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, 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, Association for Computational Linguistics, 2020, pp. 7871–7880.