International Journal of Machine Learning and Cybernetics (2024) 15:5427–5437
https://doi.org/10.1007/s13042-024-02249-6

ORIGINAL ARTICLE

Deep question generation model based on dual attention guidance

Jinhong Li[1] · Xuejie Zhang[1] · Jin Wang[1] · Xiaobing Zhou[1]

Received: 9 July 2023 / Accepted: 4 June 2024 / Published online: 19 June 2024

Abstract
Question generation refers to the automatic generation of questions by computer systems based on given paragraphs and answers, which is one of the research hotspots in natural language processing. Although previous work has made great progress, there are still some limitations: (1) The rich structural information hidden in word sequences is ignored. (2) Current studies focus on sequence-to-sequence-based neural networks to maximize the use of question-and-answer information in the context. However, the context often contains a large number of redundant and irrelevant sentences, and these models fail to filter redundant information or focus on key sentences. To address these limitations, we use a Graph Convolutional Network (GCN) and a Bidirectional Long Short Term Memory (Bi-LSTM) Network to capture the structure and sequence information of the context simultaneously. Then, we use a contrastive learning strategy for content selection to fuse the document-level and graph-level representations. We also use a dual attention mechanism for the passage and answer. Next, we use the gating mechanism to dynamically assign weights and merge them into context information to support the question decoding by modeling their interaction. We also conduct qualitative and quantitative evaluations on the HotpotQA deep question-centric dataset, and the experimental results show that the proposed model is effective.

Keywords Question generation · Bidirectional long short-term memory (Bi-LSTM) · Graph convolutional network (GCN) · Dependency graph · Attention mechanism · Contrastive learning

1 Introduction

Question Generation (QG) aims to teach machines to ask human-like questions from a range of inputs such as natural language texts [1], images [2], and knowledge bases [3]. In recent years, QG has received increasing attention due to its wide applications, which are widely applied in dialogue system [4], question answering systems [5], machine reading comprehension [6], and education systems [7].

Traditional methods for QG rely on heuristic rules or hand-crafted templates to transform a descriptive text into a related question, which can be divided into three categories, i.e., template-based [8], syntax-based [9], and semantic-based [10] methods. Generally, these methods perform two steps, i.e., context selection and question construction, to consider the answer and question types, respectively. Then, given the context with the selected topic, question construction converts the intermediate representations to a natural language question, taking either a transformation-based or template-based approach. However, such methods rely on effective hand-crafted features, often time-consuming and requiring domain-dependent expertise and experience. Moreover, they usually comprise the pipelines of several independent components with low generalizability and scalability.

Recently, most neural approaches have formulated the QG task as a Sequence-to-Sequence (Seq2Seq) problem and designed different types of encoders and decoders to improve the quality of generated questions. The first neural QG model was introduced in 2017 [1], achieving better performance than traditional rule-based approaches by employing a vanilla Recurrent Neural Network (RNN)-based Seq2Seq model with attention. Since then, a surge of follow-up enhanced models have been proposed [11–15], the majority of these models rely exclusively on Recurrent Neural Networks (RNNs) to capture sequence information of the context to generate questions. However, these methods often ignore the hidden structural information associated with a word sequence, such as the syntactic parsing relations. Thus,

✉ Xiaobing Zhou
zhouxb@ynu.edu.cn

[1] School of Information Science and Engineering, Yunnan University, Kunming 650500, Yunnan Province, China

these methods may only partially exploit the rich textual structures complementing the simple word sequence.

In recent years, Graph Neural Networks (GNNs) have achieved remarkable success in graph learning. GNNs adopt a message-passing mechanism to obtain node embeddings by aggregating and transforming the embeddings of their neighbors. Due to their strong aggregation ability, GNNs have been applied to capture the dependency structure of input passages for generating questions [16–24]. GNN-based methods can capture not only long-distance dependencies between tokens, but also enhance the model's ability to perceive the semantic structure of contexts. Many models based on GNNs have demonstrated strong capabilities in capturing the structural information of the context for QG.

Although these GNN-based studies have made remarkable progress in QG, a significant performance gap remains between machines and humans. The probable causes are threefold: First, using alone graphs pruned by Dependency Parse (DP) is insufficient to convey the total information of the passage, i.e., sequence information in the passage is not captured by the graph. Second, the DP, especially obtained by external tools, may bring an error propagation problem. Third, these GNN-based methods lose word order information and ignore the contextual relationship between sentences during the process of aggregation learning. Besides, most research on question generation focuses on simple-hop question generation, which is relevant to one fact obtainable from a single sentence, as illustrated in Fig. 1a. In real-world applications, we need the model to generate sufficiently complex and high-quality questions, requiring multi-step reasoning based on multiple information points while understanding the semantics of multiple pieces of document, as demonstrated in Fig. 1b. This more challenging task requires identifying relevant information from multiple paragraphs and reasoning over them to fulfill the generation. However, not all information in the document is equally important, and only a small portion of sentences contain key information points. Taking Fig. 1b as an example, paragraphs A and B contain key information relevant to the question. When given an answer, the model needs to accurately capture the information point "Delhi". Therefore, selecting sentences with semantic priority and ignoring invalid information points can help construct a more robust question-generation system. This approach is similar to the process of human questioning, where necessary knowledge points are extracted first (what to ask), and natural questions are constructed based on these knowledge points (how to ask).

Furthermore, some generated questions cannot be answered by the context, which is fatal to QG and also means that the generated questions are insufficient in answerability. In order to improve the relevance of questions and answers, in addition to making full use of the position of the standard answer, semantic information, and paragraph context [25, 26], existing research mainly uses the attention mechanism to dynamically extract the internal information of the paragraph [27, 28]. However, this method is the primary paragraph-level model; the global attention distribution needs to be more balanced, and the performance on long text could be better. We will focus on key sentences based on dual attention, dynamically adjusting the focus on information at different levels to reduce interference from redundant information. Additionally, we will adaptively adjust the importance of the sentence where the answer is located. In most cases, the sentence where the answer is located still occupies a central position and is directly related to the question, making it a key sentence. In other cases, using the sentence where the answer is located as a clue can also help better determine the position of key sentences, jointly forming a specific and complete semantic understanding.

It has been observed that training Seq2Seq models using cross-entropy based objectives generally has some limitations, such as exposure bias and inconsistency between train and test measurements, and may not always produce the best results on discrete evaluation metrics.

To address the above-mentioned challenges, this paper proposes a question generation model called BGA-QG, which combines Bi-LSTM, GCN, content selection strategy based on contrastive learning, and a dual guided attention mechanism. Our main contributions can be summarized as follows:

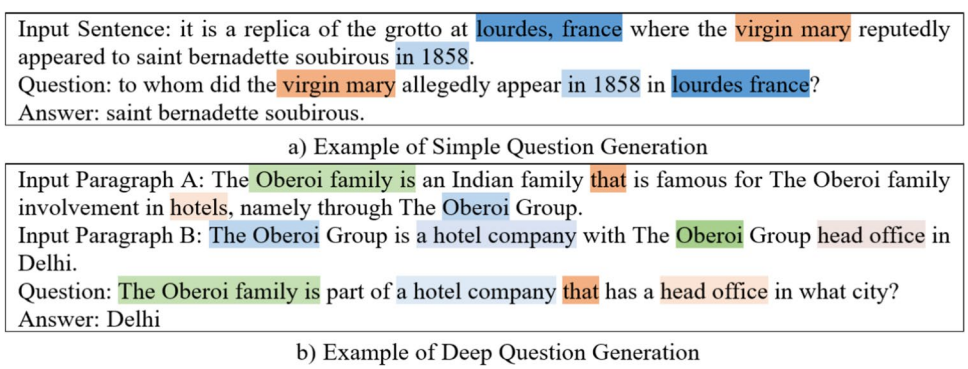

Fig. 1 Examples of simple/deep QG. The evidence needed to generate the question is highlighted

(1) We use Bi-LSTM to capture the context information of the sentence. Based on this, we conduct syntactic dependency analysis to establish the dependency matrix of the sentence. Additionally, we capture long-distance dependency information using GCNs.
(2) We introduce an auxiliary content selection task that jointly trains with question decoding. We construct positive and negative examples based on the contrastive learning method, which assists the model in selecting relevant contexts in the semantic graph to form a proper reasoning chain.
(3) We incorporate answer information into question generation and use both the passage and answer to support question decoding by modeling their interaction, implicitly completing the process of relevant context selection.
(4) Experiments demonstrate that our model performs significantly better than baseline models on the Hotpot dataset. Human evaluations indicate that our model generates questions with better grammar, relevance, and answerability.

2 Related work

In recent years, QG has attracted the attention of researchers. Most early automated question generation is based on rules or templates designed by hand to convert a given text into a problem [29–31]. Heilman et al. [29] used conversion rules to convert statements into questions and reorder multiple results to select higher-quality questions. Labutov et al. [30] first represented the original text in a low-dimensional space. Crowdsourced candidate question templates aligned with this space and finally ranked the potentially relevant templates of the new text regions to generate more profound difficulty questions. Kumar et al. [31] first represented the original text in a low-dimensional space, then crowdsourced it with candidate question templates aligned in that space, and finally ranked potentially related templates for the new text area, thus generating more complex questions. These traditional methods rely on manual labor and require professional linguistic knowledge to construct grammatical and semantic templates, which cannot adapt to flexible and changeable source text contexts. It can be challenging to extend from one domain to another.

To make up for the deficiency of rule-based methods in question generation research work. Du et al. [1] applied sequence-to-sequence models to the QG task for the first time and achieved impressive performance. Subsequently, some scholars have carried out many studies based on Seq2Seq [11, 13, 32]. Zhao et al. [13] proposed a gated self-attention encoder with answer tagging and a maxout pointer decoder, applicable to both sentence and paragraph-level inputs. Zhou et al. [11] encoded the BIO tags of answer position to real-valued vectors and then fed the answer position embeddings to the feature-rich encoder in the RNN-based Seq2Seq model. Zamani et al. [32] proposed clarifying question generation models trained via weak supervision for open-domain search queries [33]. Shi et al. [34] introduced the attention distillation module in the Occlusion-adaptive Deep Network (ODN) model to improve performance. Ma et al. [35] combined bidirectional long short-term memory (Bi-LSTM) and attention mechanism to capture the spatial-temporal dependencies.

With the proposal of dependency relations, it plays a crucial role in the field of natural language. Dependency relations, especially long-distance syntactic relations, are very useful for understanding complex sentence structures (e.g., long clauses or complex scoping). For the strong ability of aggregation, some researchers have also begun to explore the role of dependency relations in the QG task. Chen et al. [17] proposed a novel Bidirectional Gated Graph Neural Network to capture the structure information between words in a sentence. Chai et al. [18] used GNNs to interact with answers and passages to obtain new representations and fused representations. Pan et al. [19] employed an attention-based Gated GNNs model to encode semantic information of the context. Huang et al. [21] used a Graph Convolutional Network and a Bi-LSTM Network to capture the structure information and sequence information of the context. Ma et al. [22] proposed a graph-augmented sequence-to-sequence (GASeq2Seq) model, which discovered both the structure and semantic information of the passage. Shuai et al. [23] proposed an end-to-end question-distractor joint generation framework, and found that distractors are somehow relevant to the background articles by suppressing those related parts. Guan et al. [24] proposed combining reinforcement learning with semantic-rich information to generate deep questions.

In most cases, the context contains a lot of content that is not relevant to the question, and it is easy to be interfered by this redundant information when generating questions, resulting in lower-quality of generated questions. Pan et al. [19] and Su et al. [36] introduced the method of graph neural network to assist the generation of multi-hop questions. Most of the previous work is to build a graph model based on entity relationships or different semantic granularity to explicitly learn the relationship between multiple pieces of text. On this basis, we use contrastive learning to assist in selecting key information points, thereby improving the effect of multi-hop question generation. Current research mainly uses the attention mechanism to dynamically extract the information inside the paragraph. This paper is based on the double attention mechanism and uses both the passage and answer to support question decoding by modeling their interaction.

In addition, due to the continuous expansion of large-scale corpora, some pre-trained models perform significantly in QG tasks [37–40], which utilize vast amounts of unlabeled text corpora in the pre-training stage to learn general language representations and then fine-tune in supervised QG tasks. Contrastive learning [41, 42] by creating positive and negative samples for unseen (or incorrect) inputs has achieved significant success in representation learning and has recently attracted extensive attention in neural text generation.

3 Methodology

The task of question generation can be formulated as follows: Given the original text composed of different documents $D = d_1, d_2, \ldots, d_n$(contains n words) with answer $A = a_1, a_2, \ldots, a_m$ (contains m words), the purpose of QG is to generate a grammatically coherent and correct question $Q = q_1, q_2, \ldots, q_l$(contains l words).

The deep question generation model proposed in this paper employs a dual guidance attention network structure, as depicted in Fig. 2. The model comprises four key components: the context information encoding module, the dependency information encoding module, the content selection module, and the question generation module.

(1) Contextual information encoding module: Using a Bi-LSTM neural network to encode word embedding information, the context information of sentence-level text can be captured.
(2) Dependency information encoding module: By using the answer tags, POS features, the contextual information encoded as hidden state vectors in (1) as input, and constructing a graph convolutional neural network based on syntactic dependency relationships, the long-range dependency features of sentence-level text can be captured.
(3) Content selection module: To assist the model in identifying the parts worthy of questioning that form a proper reasoning chain in the semantic graph, we propose a content selection auxiliary task based on a contrastive learning strategy and train it jointly with question decoding.
(4) Question generation module: By leveraging attention mechanisms to learn contextual and dependency information, and by constructing and integrating attention information of the paragraph and the sentence where the answer is located, the focus on the sentence where the answer is located is further improved while preserving the connections between multiple sentences.

3.1 Context information encoding module

In this paper, the Bi-LSTM neural network is used to encode text information and capture contextual information about words. The word vector Glove is used to embed the text words into the feature space of d_w dimension, and the discrete word sequence is mapped to obtain the corresponding

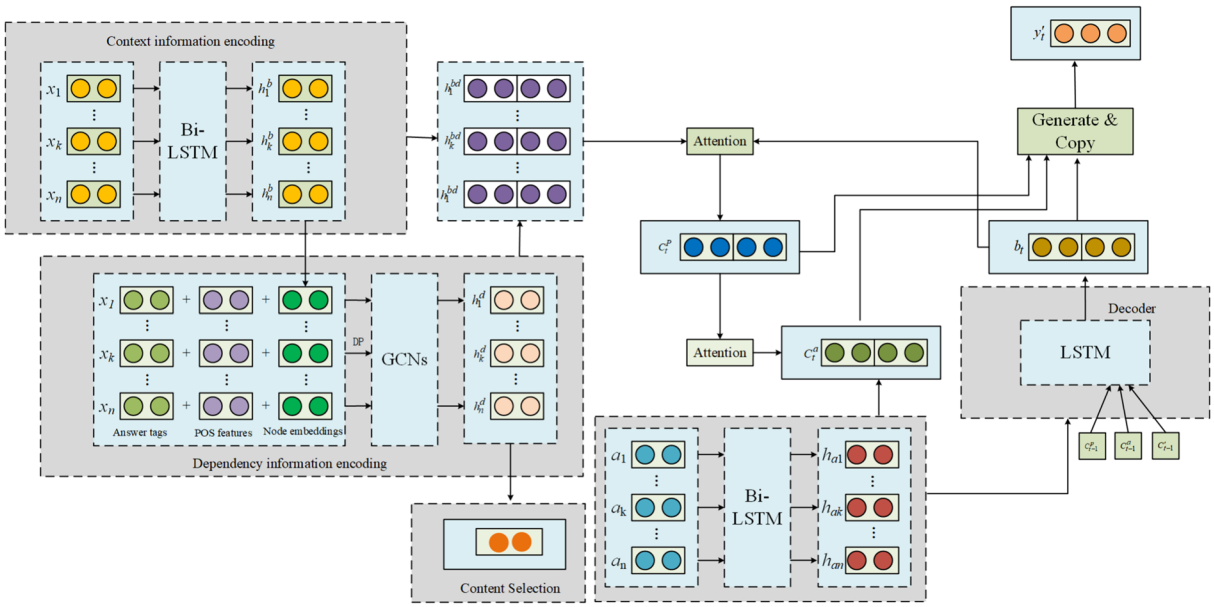

Fig. 2 Illustration of our proposed model for QG task. Bi-LSTM encoder is used to capture contextual sequence information, GCNs are used to capture contextual structure information, DP represents dependency parsing

continuous word vector representing e. Then, the word vector is put into Bi-LSTM to obtain the sentence representation h_d with contextual information:

$$\overrightarrow{h_c^d} = f_{\overrightarrow{LSTM}}([e_1, e_2, \ldots, e_n]) = [\overrightarrow{h_1^d}, \overrightarrow{h_2^d}, \ldots, \overrightarrow{h_n^d}] \tag{1}$$

$$\overleftarrow{h_c^d} = f_{\overleftarrow{LSTM}}([e_1, e_2, \ldots, e_n]) = [\overleftarrow{h_1^d}, \overleftarrow{h_2^d}, \ldots, \overleftarrow{h_n^d}] \tag{2}$$

$$h_c^d = [\overrightarrow{h_c^d}; \overleftarrow{h_c^d}] = [h_1^d, h_2^d, \ldots, h_n^d] \tag{3}$$

where $\overrightarrow{h_c^d} \in R^{n \times d}$ represents the forward hidden state vector sequence, $\overleftarrow{h_c^d} \in R^{n \times d}$ represents the backward hidden state vector sequence, $h_c^d \in R^{n \times 2d}$ represents the hidden state vector sequence output by Bi-LSTM encoding, and d is the dimension of the hidden state vector output by unidirectional LSTM.

3.2 Dependency information encoding module

In order to extract semantic information from documents, we employ dependency relationship [43] to construct a semantic graph through parsing. The resulting semantic graph of the document is a heterogeneous multi-relation graph $G = (V, E)$, where $V = (v_i)i = 1{:}N^v and E = (e_k)k = 1{:}N^e$ represent the graph nodes and the edges connecting them, where N^v and N^e are the numbers of nodes and edges in the graph, respectively. Each node v is first initialized with $v = \{w_j\}_{j=m}^n$, where w_j is the contextual representation of the word in that node and m/n is the start/end position of the text span.

First, we concatenate the last hidden states of the document encoder in both directions as the document representation h_c^d. Afterwards, for a node v, we calculate the attention distribution of h_c^don all words in v as follows:

$$r_j^v = \frac{\exp(Attn(h_c^d, w_j))}{\sum_{k=m}^v \exp(Attn(h_c^d, w_k))} \tag{4}$$

The node is initialized as $h_v^0 = \sum_{j=m}^n r_j^v w_j$. Word-to-node attention ensures each node captures the meaning of its constituting part and the semantics of the entire document. The node representation is then enhanced with two additional features: the POS embedding p_d and the answer tag embedding a_d to obtain the enhanced initial node representations. The final embedding of the i-th word in the sentence is denoted as:

$$h_d^0 = [h_v^0; p_d; a_d] \tag{5}$$

The dependency graph of any sentence can be regarded as a directed graph containing n nodes, where each node represents the corresponding word in the sentence, and each edge represents the syntactic dependency between words in the dependency graph. We employ a multi-layer GCN model (GCNs) to learn the dependencies between words by modeling the dependency graph of word sequences. GCN differs from traditional LSTM models and is an effective way to process unstructured information data. GCNs leverage dependency paths for effective information transfer and update node representations by aggregating transferred information. The dependency information between words is specifically modeled using k-order neighborhood transfer.

$$s_{ij} = W_0^T \sigma(W_0[h_{di}; h_{dj}] + b_0) \tag{6}$$

$$\beta_{ij} = \frac{\exp(s_{ij})}{\sum_{j \in N_{(i)}} \exp(s_{ij})} \tag{7}$$

$$h_{di}^{l+1} = \sigma\left(h_{di}^l + \sum_{j \in N(i)} \Lambda_{ij} \beta_{ij} (W_1 h_{dj}^l + b_1)\right) \tag{8}$$

where $N(i)$ denotes the neighbors of node i, h_{di} and h_{dj} are respectively the representations of node i and node j. β_{ij} is the attention coefficients between two nodes, σ denotes a non-linear function, and $[h_{di}; h_{dj}]$ represents the concatenation operation of h_{di} and h_{dj}, Λ_{ij} represents an element in the adjacency matrix, W_0, b_0, W_1, b_1 are learned parameters.

Finally, the output H^d of a stacked l-layer GCN can be obtained according to the following formula:

$$H^d = \{h_{di}^{l+1}\}_{i=1}^n \tag{9}$$

3.3 Content selection module

To raise a deep question, we propose an auxiliary task of content selection to jointly train with question decoding. We formulate this as a node classification task, i.e., deciding whether each node should be involved in the process of asking, i.e., appearing in the reasoning chain for raising a deep question.

To achieve this, we add one feed-forward layer on top of the final layer of the graph encoder, taking the output node representations H^d for classification. We annotate the dataset and use this supervision signal to accelerate the model training. Specifically, for each sentence in the input, the label $f_i \in \{0, 1\}$ indicates whether the ith sentence provides supporting evidence for the question-answer pair in the example. Due to the presence of labels, we employ a contrastive learning strategy where more than one sample is known to belong to the same class. However, generalizing to an arbitrary number of positives leads to a choice between multiple possible functions. In our sample, the questions and answers

that provide factual support are positive examples, while those that cannot provide support are negative examples.

We consider a node to be positive ground-truth for the content selection task if its contents appear in the ground-truth question. Content selection helps the model identify the parts of the text most relevant to the question and form a coherent reasoning chain in the semantic graph. We train these two tasks jointly and share the weights of the input representations.

3.4 Question generation module

The attention mechanism dynamically adjusts weight parameters based on the decoding state of the sequence, enabling it to capture context information that is most relevant to the current decoding moment. The main idea is to calculate attention weights or scores between the decoder's hidden state and all the encoder's hidden states at the current time step, and assign weights to each encoder's hidden state accordingly. In this paper, we calculate attention on the semantic representation of both the text and answer, which facilitates dynamic attention to different levels of information, adaptive adjustment of the importance of the sentence where the answer is located, and reduction of distracting information from irrelevant parts of the text. The calculation process is illustrated as follows:

$$\alpha_t^d = soft\max(H^d W_d u_t) \tag{10}$$

$$c_t^d = \sum_{i=1}^{m} \alpha_{t,i}^d H_i^d \tag{11}$$

where W_d stands for a learning matrix, u_t represents the hidden state of the decoder at time t, m represents the hidden state of the decoder at time t, a_t^d represents the normalized attention score, c_t^d is the weighted sum of the all word representations in passage, that contains the most relevant information from the passage representation.

Afterward, to better utilize the information from both the passage and the answer, we use the contextual representation of the passage to further extract the important information from the answer. The calculation process is shown as follows:

$$\alpha_t^s = soft\max(H^a W_a C_t^d) \tag{12}$$

$$c_t^a = \sum_{j=1}^{n} \alpha_{t,j}^s H_j^a \tag{13}$$

where W_a stands for a learning matrix, a_t^s represents the normalized attention score, c_t^a represents the weighted sum of all word representations in answer.

In order to obtain a context vector required for the final generation stage, two levels of context vectors need to be combined using one of three fusion modes: addition, splicing, or gating mechanism fusion mode. The sentence further explains that the addition method directly adds the paragraph and sentence-level context vectors to obtain the fusion context vector.

$$c_t = c_t^d + c_t^a \tag{14}$$

As for the fusion method of splicing, this paper first splices the paragraph-level and sentence-level context vectors, and splices the fusion vector obtained by multiplying the two. The splice vectors are then passed through a multilayer perceptron transform to achieve fusion processing.

$$c_{concat} = [c_t^d; c_t^a; c_t^d \odot c_t^a] \tag{15}$$

$$c_t = \tanh(W^c c_{concat} + b) \tag{16}$$

where $\odot$ represents the hadamard product, the product of corresponding elements in two vectors. W^c is the trainable model parameter and b is the bias term.

As for the fusion mode of the gating mechanism, its main feature is that the hidden state of the decoder is considered again. In this paper, the hidden state of the decoder and the obtained paragraph-level and sentence-level context vectors are firstly calculated into a gated score, as shown in Eq. (17).

$$g_t = \sigma(W^g [s_t; c_t^d; c_t^a]) \tag{17}$$

where W^g is a trainable model parameter representing the sigmoid function, this score is used to assign weight between two levels of attention, enabling the pattern to focus dynamically on different levels of contextual information, as shown in Eq. (18).

$$c_t = g_t c_t^d + (1 - g_t) c_t^a \tag{18}$$

After the fusion of dual attention, the fusion context vector c_t is obtained and input to the decoding layer for question generation.

For the question decoding, we employ an LSTM as the decoder to realize the process of word-by-word generation conditioned fusion context vector, and the representations of the previously generated words.

$$u_t = LSTM(u_{t-1}, c_{t-1}, y_{t-1}) \tag{19}$$

where u_{t-1} represents the hidden state of LSTM at time $t-1$, y_{t-1} represents the word generated by the decoder at time $t-1$. When $t=0$, $u_0 = H^d$ stands for the decoder's initial state.

Finally, the fused contextual vector obtained from the dual attention layer is projected onto the word distribution vector to obtain the probability of the current decoding time step's corresponding word. The calculation process is as follows:

$$d_t = \tanh(W_v[u_t;c_t]) \tag{20}$$

$$P_{voc} = soft\max(W_o d_t) \tag{21}$$

W_v and W_o are trainable model parameters, and P_{voc} represents a vector whose dimension is a fixed vocabulary. For new words outside the restricted vocabulary, i.e., unregistered words, the model can only generate the universal tag $< unk >$, and the question may lack essential words.

To solve the problem of rare words, the decoder applies a copy mechanism. we directly leverage row attention scores α_t^d as the score of the copy mechanism scoreP_{copy}. Finally, the probability distribution of each word at the t step can be calculated as follows:

$$p\left(y_t \mid \{y_{<t}\}\right)_{\text{final}} = \text{soft max}\left(\left[P_{voc};P_{copy}\right]\right) \tag{22}$$

4 Experiments

4.1 Dataset

To assess the performance of our proposed method in the question generation task, we conduct experiments using the Hotpot dataset. Hotpot is a collection of QA data comprising over 113,000 question-answer pairs sourced from Wikipedia articles. Each question is accompanied by two supporting documents that provide the necessary evidence for inferring the answer. The dataset consists of two primary types: comparison and bridge. For data preprocessing, we adhere to the original data split of Hotpot, with 90,440 samples for training and 7045 samples for evaluation. For a more accurate assessment of the model's generalization ability, we take 1000 samples from the 90,440 samples for validation, and the remaining samples for training. The data in the validation set is completely independent of the training set, ensuring a fair evaluation of the model's performance on unseen data.

4.2 Evaluation metrics

4.2.1 Automatic evaluation

This paper uses the following three automatic evaluation indicators to evaluate the effectiveness of the question-generation method:

BLEU [44] is a commonly used evaluation method for automatic translation. It evaluates the adequacy and fluency of generated sentences by counting the number of matching segments between generated sentences and standard sentences. BLEU-1, BLEU-2, BLEU-3, and BLEU-4 use 1-gram to 4-gram for calculation, respectively.

METEOR(ME) [45] is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision. Its purpose is to improve some inherent shortcomings in the BLEU standard. Additionally, it considers some functions not found in other indicators, such as synonym matching, so it is farther away from the effect of manual evaluation.

ROUGE-L (R-L) [46] measures recall by assessing how much the words in reference sentences appear in predictions using the longest common subsequence-based statistics. ROUGE is an indicator calculated using a recall-based similarity measure. Its basic idea is to use the n-tuple co-occurrence probability of the model-generated text and the reference text as the basis for evaluation. However, it cannot evaluate whether the sentence is fluent. In the paper's evaluation, ROUGE-L is selected as the evaluation standard, and its basic idea is to match the most extended typical sequence between two text units.

4.2.2 Human evaluation

In order to more accurately evaluate the performance of the problem generation model, we randomly select 100 samples from the test set and conduct a manual evaluation. We compare our model with SG_DQG, MultiQG, Ass2s-a, and NQG++, where annotators evaluate the generation quality from three important aspects of deep questions: fluency, complexity and answerability. The evaluation criteria are as follows:

(1) Fluency: Whether the generated questions are naturally fluent in terms of grammar and semantics, scoring from 1 to 5.
(2) Complexity: Whether the generated questions require two or more information points to answer reasoning questions, with scores ranging from 1 to 5.
(3) Answerability: Whether the generated question can be answered from the document and is consistent with the given answer, the score is 0 or 1.

4.3 Experiments settings

In order to extract semantic information from documents, we use the dependency parsing method to construct a semantics graph. The maximum stage length of the original text, the answer and the question is 512,10,50 respectively. We utilize pre-trained Glove word embeddings, whose dimension

Table 1 Performance comparison with baselines

Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
Seq2Seq+Attn [47]	32.97	21.11	15.41	11.81	18.19	33.48
NQG++ [11]	35.31	22.12	15.53	11.50	16.96	32.01
ASs2s [12]	34.60	22.77	15.21	11.29	16.78	32.88
MP-GSA [13]	34.38	23.00	17.05	13.48	18.39	34.51
MulQG [16]	40.08	26.58	19.61	15.11	20.24	35.35
QG-Reward [15]	37.97	–	–	15.41	19.61	31.85
SG_DQG [19]	40.55	27.21	20.13	15.53	20.15	36.94
ADDQG [20]	**44.34**	**31.32**	**22.68**	**17.54**	20.56	38.09
BGAC-QG(ours)	41.32	28.12	21.03	16.22	**21.02**	**38.42**

The best performance is in bold

is set to 300. The dimensions of Answer tags, POS tags, and NER tags are set to 100, 60, and 50, respectively. The hidden size of LSTM is 300 for all encoders and decoders. For the GCN encoder, the level of the stacked layer is 4. We make the vocabulary with the top 45,000 frequency words. Optimization is performed by Adam, with an initial learning rate of 0.0025, and the weight decline rate is 0.001. For the decoder part, we use beam search with a beam size 12 to get the final result.

4.4 Results and analysis

4.4.1 Baselines

We compare our proposed model against several strong baselines on question generation.

(1) Seq2Seq + Attn [47]: It is a Seq2Seq model with the attention mechanism. We connect the document with the answer as the input of the encoder.
(2) NQG++ [11]: It introduces rich linguistic features into the encoder, including entity information, answer location and POS tagging.
(3) Ass2s [12]: It proposes an answer-separated Seq2Seq model by replacing the answer in the input sequence with some specific words.
(4) MP-GSA [13]: It proposes a gated attention mechanism and a maximum pointer to improve document-level problem generation.
(5) SG_DQG [19]: It constructs a semantic graph neural network to improve the generation of multi-hop problems.
(6) QG-Reward [15]: It introduces a solid graph neural network based on GCN into the Seq2seq model.
(7) MulQG [16]: It introduces a GCN-based entity graph neural network into the Seq2seq model.
(8) ADDQG [20]: It proposes an Answer-driven Deep Question Generation model based on the encoder-decoder framework.

Table 1 presents the experimental evaluation results of the BGAC-QG model and various baseline models on the HotpotQA dataset. The results indicate that the BGAC-QG model outperforms all baselines in terms of METEOR and ROUGE-L scores. Specifically, compared to the previous state-of-the-art model ADDQG, the BGAC-QG model achieves improvements of 0.46 and 0.33% in METEOR and ROUGE-L scores, respectively. This improvement can be attributed to the utilization of contrastive learning in the BGAC-QG model, which enables effective extraction of key information from different documents and filtering out irrelevant information, resulting in higher-quality generated questions. However, our model is slightly worse than ADDQG regarding the BLEU metric. The reason for this is that ADDQG employs 7 attention mechanisms to capture the key information in the document and fine-tunes the model using reinforcement learning. While this approach facilitates the capture of consecutive words in the document to some extent, it often requires extensive training iterations to achieve satisfactory performance, especially given the large question space in the HotpotQA dataset, leading to longer training times.

In addition, our model's other metrics surpass those of the baseline model, demonstrating the significant impact

Table 2 Results of ablation

Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METOR	ROUGE-L
BASE	36.48	20.56	12.89	8.46	15.43	30.86
W/GCNs	37.63	24.81	18.14	13.85	19.24	34.93
W/CL	39.44	26.23	19.26	14.20	19.84	36.14
W/DA	**39.46**	**26.34**	**19.35**	**14.36**	**20.12**	**36.32**

The best performance is in bold

Table 3 Results of human evaluation

Models	Fluency	Complexity	Answerability
BGAC-QG	**4.55**	**4.25**	**0.78**
SG_DQG	4.36	3.95	0.71
MultiQG	4.27	4.04	0.76
Ass2s-a	4.02	2.43	0.65
NQG	3.25	2.46	0.63

The best performance is in bold

of semantically rich document representation on the generation of deep questions, and the crucial role played by auxiliary content selection. Specifically, the SG_DQG and BGAC-QG models are based on semantically rich document representation for question generation, with variations in how answers and documents are encoded. Notably, our BGAC-QG model achieves average improvements of 0.69, 0.87, and 1.48% in BLEU-4, METEOR, and ROUGE-L scores, respectively, showing superior performance in deep question generation through contrastive learning-based text selection.

Furthermore, BGAC-QG demonstrates significant superiority over the Seq2Seq+Attn and NQG models in terms of performance, thereby highlighting the limitations of the Seq2Seq+Attn and NQG models in the generation of deep questions.

4.4.2 Ablation study

In order to thoroughly analyze the impact of each module on the experimental results, this paper conducted ablation experiments from several aspects, including the impact of the context information encoding module, dependency information encoding module, text selection, and attentional fusion methods. The results of these experiments are presented in Table 2.

According to the results of the comparison experiment, we can conclude that the model based on GCN is generally superior to the model constructed solely using LSTM. This suggests that GCN can learn information other than contextual semantic information to assist in sentiment analysis of the target. Specifically, LSTM focuses more on the contextual information of words, and with the increase in sentence length, noise information irrelevant to the sentiment analysis of the target also increases. However, by constructing graphs, GCN can achieve long-distance connections and information propagation between words. Additionally, by using the syntax dependency tree to construct the graph, it is possible to further learn the dependency relationship between words and obtain the final representation of the target based on the dependency relationship.

When the content selection task is turned off, from Tables 1 and 2, it can be seen that the BLEU-4 score drops from 16.22 to 14.20, indicating the contribution of joint training with the auxiliary task of content selection. Moreover, content selection helps train the model to focus on question-worthy content and form a correct reasoning chain in question decoding. Without the intent classifier, the model tends to generate non-differential questions, which may significantly decrease the performance of both relevance and diversity. This reveals that the intent of the question provides not only auxiliary information to generate meaningful questions but also plays a vital role in making the questions from the same passage distinguishable.

Regarding the dual attention fusion method, we are considering changing it to the cancellation fusion, addition fusion, and concatenation fusion methods. From the table, it can be seen that the gating mechanism fusion method can obtain the best model performance, indicating that the model still requires the participation of decoder hidden states during attention fusion to dynamically adjust the focus on attention information at different levels and achieve more reasonable weight allocation.

4.4.3 Human evaluation

Table 3 shows our human evaluation results, further validating that our model generates questions of better quality than the baselines. Let us explain two observations in detail: Compared to the baseline model, our model shows improvements in fluency, complexity, and relevance. This is because

Fig. 3 Generated question cases from different models

Input Paragraph A: Kansas Song (We're from Kansas) is a fight song of the university of Kansas. The main campus in Lawrence, one of the largest college towns in Kansas, is on Mount Oread, the highest elevation in Lawrence. Input Paragraph B: Two branch campuses are in the Kansas City metropolitan area: the Edwards Campus in Overland Park, and the university 's medical school and hospital in Kansas City. Answer: Kansas Song.
Ground truth: What is the name of the fight song of the university whose main campus is in Lawrence? Seq2Seq: What is the name of the fight song of the university? QG-Reward: What is the name of the fight song of the university in Lawrence? SG_DQG: What is the name of the fight song whose main campus in Lawrence? Ours: What is the name of the fight song of the university whose main campus is in Lawrence, Kansas and whose branch campuses are in the Kansas City metropolitan area?

the baseline model tends to generate more simplistic questions (which affects complexity) or questions with semantic errors (which affects fluency). By incorporating the semantic graph, our model can generate questions with fewer semantic errors and leverage more contextual information. The model can enhance the performance of generating multi-hop questions by identifying and capturing relevant information points for questioning and conducting accurate reasoning.

4.4.4 Case study

We present some examples of generated questions in Fig. 3. The Seq2Seq model lacks the ability to capture contextual structure information, leading to its failure to learn that "university" refers to"whose main campus is in Lawrence". The QG-Reward model does not make full use of the sequence information between words, resulting in the generated questions being inconsistent with the context facts and unable to be answered by the given answer "Kansas Song". As for SG_DQG, although it generates questions without grammatical errors, it cannot be answered by given answers. In contrast, the questions generated by our model are more specific, and with better answerability and grammaticality.

4.5 Conclusion

This paper proposes an automatic question-generation model for deep question-generation tasks, This model utilizes a contrastive learning strategy to represent each sentence as a node on a graph and learns the semantic relationship between nodes using graph attention neural networks. Based on the hidden state representation of the graph, the model can predict the most important sentences for generating the answer to a given question, with the sentence containing the answer as the primary clue. To further improve the model's accuracy, the model uses dynamic fusion and allocation of dual attention to focus on key sentences and reduce the interference of redundant information. Experimental results demonstrate that this model effectively improves the performance of deep question generation.

Funding This work was supported by the Natural Science Foundation of China under Grants No. 62266051 and No. 61966038.

Data availability No new data were created during the study.

Declarations

Conflict of interest There are no conflicting interests known to the authors.

References

1. Du X, Shao J, Cardie C (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), pp. 1342–1352
2. Patel A, Bindal A, Kotek H, Klein C, Williams J (2021) Generating natural questions from images for multimodal assistants. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2270–2274
3. Serban IV, García-Durán A, Gulcehre C, Ahn S, Chandar S, Courville A, Bengio Y (2016) Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 588–598
4. Wang Y, Liu C, Huang M, Nie L (2018) Learning to ask questions in open-domain conversational systems with typed decoders. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2193–2203
5. Tang D, Duan N, Qin T, Yan Z, Zhou M (2017) Question answering and question generation as dual tasks. arXiv preprint arXiv:1706.02027
6. Danon G, Last M (2017) A syntactic approach to domain-specific automatic question generation. arXiv preprint arXiv:1712.09827
7. Yao K, Zhang L, Luo T, Tao L, Wu Y (2018) Teaching machines to ask questions. In: IJCAI, pp. 4546–4552
8. Mostow J, Chen W (2009) Generating instruction automatically for the reading strategy of self-questioning. In: Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling. pp. 465–472
9. Kunichika H, Katayama T, Hirashima T, Takeuchi A (2004) Automated question generation methods for intelligent english learning systems and its evaluation. In: Proc. of ICCE, vol. 670
10. Huang Y, He L (2016) Automatic generation of short answer questions for reading comprehension assessment. Nat Lang Eng 22(3):457–489
11. Zhou Q, Yang N, Wei F, Tan C, Bao H, Zhou M (2018) Neural question generation from text: a preliminary study. In: Natural Language Processing and Chinese Computing: 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings 6, pp. 662–671
12. Kim Y, Lee H, Shin J, Jung K (2019) Improving neural question generation using answer separation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6602–6609
13. Zhao Y, Ni X, Ding Y, Ke Q (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, pp. 3901–3910
14. Liu B, Zhao M, Niu D, Lai K, He Y, Wei H, Xu Y (2019) Learning to generate questions by learningwhat not to generate. In: The World Wide Web Conference, pp. 1106–1118
15. Xie Y, Pan L, Wang D, Kan M-Y, Feng Y (2020) Exploring question-specific rewards for generating deep questions. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 2534–2546
16. Qiu L, Xiao Y, Qu Y, Zhou H, Li L, Zhang W, Yu Y (2019) Dynamically fused graph network for multi-hop reasoning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6140–6150
17. Chen Y, Wu L, Zaki MJ (2020) Reinforcement learning based graph-to-sequence model for natural question generation. In: 8th International Conference on Learning Representations

18. Chai Z, Wan X (2020) Learning to ask more: Semi-autoregressive sequential question generation under dual-graph interaction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 225–237
19. Pan L, Xie Y, Feng Y, Chua T-S, Kan M-Y (2020) Semantic graphs for generating deep questions. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1463–1475
20. Wang L, Xu Z, Lin Z, Zheng H, Shen Y (2020) Answer-driven deep question generation based on reinforcement learning. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 5159–5170
21. Huang Q, Fu M, Mo L, Cai Y, Xu J, Li P, Li Q, Leung H-f (2021) Entity guided question generation with contextual structure and sequence information capturing. In: Proceedings of the AAAI Conference on Artificial Intelligence, 35, pp. 13064–13072
22. Ma H, Wang J, Lin H, Xu B (2023) Graph augmented sequence-to-sequence model for neural question generation. Appl Intell 53(11):14628–14644
23. Shuai P, Li L, Liu S, Shen J (2023) Qdg: a unified model for automatic question-distractor pairs generation. Appl Intell 53(7):8275–8285
24. Guan M, Mondal SK, Dai H-N, Bao H (2023) Reinforcement learning-driven deep question generation with rich semantics. Inf Process Manag 60(2):103232
25. Zhou W, Zhang M, Wu Y (2019) Question-type driven 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), pp. 6032–6037
26. Zi K, Sun X, Cao Y, Wang S, Feng X, Ma Z, Cao C (2019) Answer-focused and position-aware neural network for transfer learning in question generation. In: Knowledge Science, Engineering and Management: 12th International Conference, KSEM 2019, Athens, Greece, August 28–30, 2019, Proceedings, Part II 12, pp. 339–352
27. Jia X, Zhou W, Sun X, Wu Y (2020) How to ask good questions? try to leverage paraphrases. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6130–6140
28. Song L, Wang Z, Hamza W, Zhang Y, Gildea D (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, Volume 2 (Short Papers), pp. 569–574
29. Heilman M, Smith NA (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, pp. 609–617
30. Labutov I, Basu S, Vanderwende L (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), pp. 889–898
31. Kumar V, Boorla K, Meena Y, Ramakrishnan G, Li Y-F (2018) Automating reading comprehension by generating question and answer pairs. In: Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3–6, 2018, Proceedings, Part III 22, pp. 335–348
32. Zamani H, Dumais S, Craswell N, Bennett P, Lueck G (2020) Generating clarifying questions for information retrieval. In: Proceedings of the Web Conference 2020, pp. 418–428
33. Li J, Gao Y, Bing L, King I, Lyu MR (2019) 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), pp. 3216–3226
34. Sadiq M, Shi D, Guo M, Cheng X (2019) Facial landmark detection via attention-adaptive deep network. IEEE Access 7:181041–181050
35. Ma J, Jia C, Yang X, Cheng X, Li W, Zhang C (2020) A data-driven approach for collision risk early warning in vessel encounter situations using attention-bilstm. IEEE Access 8:188771–188783
36. Su D, Xu Y, Dai W, Ji Z, Yu T, Fung P (2020) Multi-hop question generation with graph convolutional network. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4636–4647
37. Su D, Xu P, Fung P (2022) Qa4qg: Using question answering to constrain multi-hop question generation. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8232–8236
38. Tang C, Zhang H, Loakman T, Lin C, Guerin F (2023) Enhancing dialogue generation via dynamic graph knowledge aggregation. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 4604–4616
39. Matsumori S, Okuoka K, Shibata R, Inoue M, Fukuchi Y, Imai M (2023) Mask and cloze: automatic open cloze question generation using a masked language model. IEEE Access 11:9835–9850
40. Muse H, Bulathwela S, Yilmaz E (2023) Pre-training with scientific text improves educational question generation (student abstract). In: Proceedings of the AAAI Conference on Artificial Intelligence, 37, pp. 16288–16289
41. Zhang C, Chen Y, Liu L, Liu Q, Zhou X (2022) Hico: Hierarchical contrastive learning for ultrasound video model pretraining. In: Proceedings of the Asian Conference on Computer Vision, pp. 229–246
42. An C, Feng J, Lv K, Kong L, Qiu X, Huang X (2022) Cont: Contrastive neural text generation. In: Advances in Neural Information Processing Systems 35, pp. 2197–2210
43. De Marneffe M-C, Dozat T, Silveira N, Haverinen K, Ginter F, Nivre J, Manning CD (2014) Universal Stanford dependencies: a cross-linguistic typology. In: LREC, vol. 14, pp. 4585–4592
44. Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318
45. Denkowski M, Lavie A (2014) Meteor universal: language specific translation evaluation for any target language. In: Proceedings of the Ninth Workshop on Statistical Machine Translation, pp. 376–380
46. Lin C-Y (2004) Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81
47. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations