Enhancing Question Generation with Syntactic Details and Multi-Level Attention Mechanism

1st Cong Zhou
Zhejiang Normal University
School of Computer Science and Technology
Jinhua, China
zhoucong@zjnu.edu.cn

2nd Jia Zhu*
Zhejiang Normal University
College of Education
Jinhua, China
jiazhu@zjnu.edu.cn

3rd Qing Wang
Zhejiang Normal University
School of Computer Science and Technology
Jinhua, China
wq2481@zjnu.edu.cn

4th Chaojun Meng
Zhejiang Normal University
School of Computer Science and Technology
Jinhua, China
mengchaojun@zjnu.edu.cn

5th Changfan Pan
Zhejiang Normal University
School of Computer Science and Technology
Jinhua, China
changfanpan@zjnu.edu.cn

6th Jianyang Shi
Zhejiang Normal University
College of Education
Jinhua, China
shijianyang@zjnu.edu.cn

2023 7th Asian Conference on Artificial Intelligence Technology (ACAIT) | 979-8-3503-5914-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ACAIT60137.2023.10528429

***Abstract*—Generating questions is a pervasive task in natural language generation that utilizes sequence-to-sequence models based on recurrent neural networks. However, these models face a significant challenge known as the "long-term dependence" problem, which hinders their ability to capture long-term dependencies in sequences effectively. Therefore, the generated questions may not be related to the original answers or follow the correct grammar rules, making them less fluent. This paper proposes a question-generation model that combines syntactic details with an enhanced multi-level attention mechanism. Incorporating syntactic information vectors into the model's input and using a multi-level attention mechanism to capture sequential contextual information effectively alleviates the above problems. Experimental evaluations show that the proposed model outperforms commonly used models on the SQuAD dataset.**

***Index Terms*—Question Generation, Syntactic Information, Attention Mechanisms, Recurrent Neural Networks**

I. Introduction

Question generation is challenging in natural language processing, which involves generating semantically and syntactically correct questions from a given context [1]. Question-generation tasks are widely used and can be applied in fields such as automatic consultation, intelligent dialogue, and educational guidance. Especially in educational guidance, generating high-quality reading comprehension questions can guide learners to learn and improve their reading quality effectively. In the existing experimental scenario based on the SQuAD [2] dataset, the input of the question generation system is a declarative sentence containing the target answer, and the output is a target interrogative sentence for the target answer. Example 1 gives a data sample for the question generation task in this dataset.

There are two types of methods for generating questions from text: rule-based and neural network-based methods. Rule-based methods rely on a predefined set of transformation rules or templates manually created based on the properties of natural language to generate questions. On the other hand, neural network-based approaches leverage sequence-to-sequence architectures to encode and decode a given sentence and generate relevant questions. Researchers have successively proposed methods such as attention mechanism, copy mechanism [3], and pointer network [4] based on end-to-end models to improve the performance of generative models. However, end-to-end based generation methods still need some fixing, mainly: (1) The generated questions are not strongly correlated with the original text answers. (2) The generated sentences are not fluent and sometimes have grammatical problems.

TABLE I: Example of question generation

Input text	...In 1979, the unit was renamed as abc Motion Pictures but was later dissolved in1985....
Answer	1985
Standard question	When was the abc motion pictures eventually dissolved?

Considering the above two shortcomings in the existing research, a question generation model with an improved attention mechanism is proposed. The main improvements of this model include two aspects: First, it utilizes a parser to parse the input sentence to extract syntactic information from it. The syntactic information vector, named entity recognition vector, and part-of-speech tagging vector are concatenated to form a comprehensive vocabulary feature vector. Then the pooling operation (maximum pooling) can be used to reduce the dimensionality of the feature vectors to obtain the final vocabulary feature representation. Combine the attention mechanism to help the encoder better capture the context and syntax information in the input sequence, generating better quality and fluency questions. A multi-level self-attention mechanism has been integrated into the encoder to enhance the relevance of questions and answers(Q&A), enhancing

*Corresponding author

979-8-3503-5914-5/23/$31.00 ©2023 IEEE

the decoder's semantic understanding and generation accuracy. This method improves the relevance of the generated Q&A pairs. This article utilizes the SQuAD dataset to assess the upgraded model's performance. The experiment outcomes demonstrate that the enhanced model surpasses the presently prevalent models in automatic evaluation.

II. Related Work

Researchers are actively investigating methods to generate high-quality and fluent questions. There are generally two approaches to text-based question-generation techniques: rule-based methods and neural network-based methods.

The rule-based question generation method generates questions through predefined rules and templates. Human experts or linguists write these rules and templates, which can be tuned for specific tasks or domains. Heilman et al. [5] converted declarative sentences into question sentences through a series of conversion rules and reordered multiple results to select high-quality questions. Rule-based question generation methods rely on manual work and require researchers to have deep linguistic knowledge, so they need more flexibility and scalability.

Neural network-based question generation methods refer to methods that use deep learning models to learn to generate questions automatically. These models are usually composed of neural networks and can automatically learn the regularities and patterns that generate problems with the help of large amounts of data. Various techniques have been introduced to generate questions from text due to the swift progress of deep learning. Among these approaches is the sequence-to-sequence model suggested by Du et al. [6], which has proven effective. This model first encodes declarative sentences and then decodes the final interrogative sentences, achieving promising results in question generation. An alternative approach proposed by Song et al. [7] involves the semantic matching of answers to text so that the encoded context vector can contain the relevance of the answers. Tuan et al. [8] obtain more distant semantically relevant content for question generation through answer-independent encoding and a multi-layer attention mechanism. In addition, Chan et al. [9] modified the masking strategy of the pre-trained model BERT to make it applicable to the problem generation task and achieved excellent performance.

III. Method

The task of generating questions can be formulated as follows: given an article $P = (p_1, p_2, \ldots, p_{m-1}, p_m)$ of length n and an answer $A = (a_1, a_2, \ldots, a_{s-1}, a_s)$, the goal is to generate a question $\widetilde{Q}$ that is most relevant to the answer, as shown in formula (1).

$$\widetilde{Q} = \operatorname{argmax} \mathrm{P}(Q \mid P, A) \tag{1}$$

The question generation model utilized in this study employs an encoder-decoder framework that incorporates attention and feature fusion mechanisms. The model's architecture is illustrated in Figure 1, with the left side representing syntax tree information extraction, the middle consisting of the encoding stage encompassing three layers, and the right side corresponding to the decoding stage.

In the encoding layer, the encoder utilized in this paper is a bidirectional long-short-term memory network (BiLSTM) responsible for converting the words and their associated information from the input sentence into corresponding vectors. These vectors are then fed into the encoder to model the sentence. Typically, this is done with external tools. This paper initializes the word vectors using pre-trained embeddings from CloVe [10]. If Glove does not cover the word, its word vector is randomly initialized. The lexical feature is the splicing operation of the syntactic information vector, the named entity recognition vector, and the part-of-speech tagging vector to form a comprehensive lexical feature vector. The final vocabulary feature representation is obtained using the pooling operation (maximum pooling) to reduce the dimensionality of the feature vectors. The BIO notation is used for the position identification of the answer. This paper employs the B-I-O labeling scheme to indicate the starting position and continuation of the answer, with B representing the beginning, I representing the continuation, and O indicating any irrelevant text.

The model encodes these vectors through BiLSTM to obtain the hidden state of each word, which has prominent temporal characteristics and contains contextual semantic information. The calculation process is shown in formula (2)˜(4).

$$\overrightarrow{\boldsymbol{h}_t} = \mathrm{BiLSTM}\left(\boldsymbol{e}_t, \overrightarrow{\boldsymbol{h}_{t-1}}\right) \tag{2}$$

$$\overleftarrow{\boldsymbol{h}_t} = \mathrm{BiLSTM}\left(\boldsymbol{e}_t, \overleftarrow{\boldsymbol{h}_{t-1}}\right) \tag{3}$$

$$\boldsymbol{h}_t = \left[\overrightarrow{\boldsymbol{h}_t}; \overleftarrow{\boldsymbol{h}_t}\right] \tag{4}$$

Among them, $\overrightarrow{\boldsymbol{h}_t}$ represents the hidden state of the forward BiLSTM network, while $\overleftarrow{\boldsymbol{h}_t}$ represents the hidden state of the reverse BiLSTM network. To generate the semantic representation vector $boldsymbolh_t$ of words, the hidden states from both directions are combined by splicing. For each word in the paragraph P, the above encoding operation will be repeated in the order before and after, and the semantic representation of the paragraph obtained is $\boldsymbol{H}^P = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_m]$ The semantic representation of the answer sentence extracted from the paragraph semantics is $\boldsymbol{H}^A = [\boldsymbol{h}_1, \ \boldsymbol{h}_2, \cdots, \boldsymbol{h}_s]$.Furthermore, instead of encoding the entire sentence containing the answer, this paper extracts the pertinent information of the answer from the semantic representation of the paragraph.

The attention mechanism is a technique used in machine learning to assign weights to each input unit so that the model can concentrate on the most pertinent ones when making predictions. The model can establish more robust connections between the query and the relevant input units, leading to better performance. This paper uses the global attention score calculation method to compute sentence-level and paragraph-level attention scores. Specifically, we utilize the context

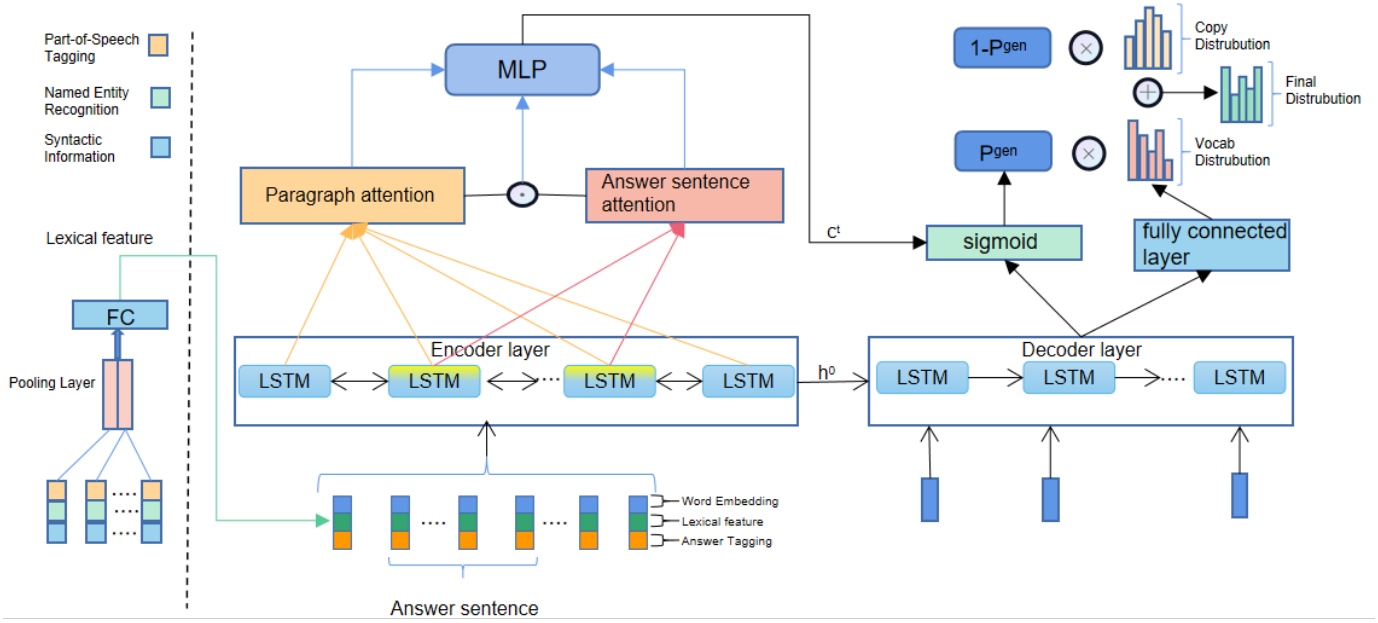

Fig. 1: Question Generation Model Framework.

vector $\boldsymbol{c}_t$, which is obtained from the encoder, along with the decoder state $\boldsymbol{s}_t$, and merge it with the encoder output $\boldsymbol{h}_i$. This process enables us to efficiently extract the pertinent details from the input sequence and produce precise forecasts. The attention calculation formula is shown in formula (5)˜(7).

$$o_{t,i} = \tanh\left(W_s s_t + W_h h_i + b\right) \tag{5}$$

$$a_{t,i} = \frac{\exp\left(o_{t,i}\right)}{\sum_{j=1}^{m} \exp\left(o_{t,i}\right)} \tag{6}$$

$$\boldsymbol{c}_t = \sum_{i=1}^{m} a_{t,i} \boldsymbol{H}_i \tag{7}$$

Among them, W_s and W_h are trainable model parameters, $o_{t,i}$ represents the attention score, and then use formulas (5) and (6) to calculate the normalized attention score at time t $a_{t,i}$, the vector $\boldsymbol{c}_t$ stands for the context at time t. Concatenate the paragraph-level and sentence-level context vectors, and concatenate the fusion vector obtained by multiplying the two. Then, the concatenated vectors are transformed by a multi-layer perceptron for fusion processing.

During decoding, the encoder's final hidden state is utilized as the initial hidden state for the decoder. At each decoding step, the previous decoded word and the preceding step's hidden state are fed as inputs. The LSTM is applied to generate the current decoder hidden state $\boldsymbol{s}_t$. Next, the fully connected layer inputs $\boldsymbol{s}_t$ and the context vector $\boldsymbol{c}_t$. The activation function is the tanh, which calculates $\boldsymbol{f}_t$. Subsequently, the softmax function normalizes $\boldsymbol{f}t$, resulting in the probability Pvocab for every word in the vocabulary, as demonstrated in equations (8) to (10).

$$\boldsymbol{s}_t = \mathrm{LSTM}\left(\boldsymbol{y}_{t-1}, \boldsymbol{s}_{t-1}\right) \tag{8}$$

$$\boldsymbol{f}_t = \tanh(W[\boldsymbol{s}_t, \boldsymbol{c}_t] + b) \tag{9}$$

$$P_{\text{vocab}} = \mathrm{softmax}\left(\boldsymbol{f}_t\right) \tag{10}$$

Considering the limited vocabulary size, generating question words only through P_{vocab} will be negatively affected by rare words and unknown words, resulting in low-quality final generated questions. The decoder has been enhanced with a coping mechanism to selectively extract relevant words from the input language and merge them into the generated problem to solve this problem. This is done using the previously computed global attention weights as the probability distribution of the copied words P_{copy}. The input word vector $\boldsymbol{y}_t$, the hidden state $\boldsymbol{s}_t$, and the context vector $\boldsymbol{c}t$ of the decoder are then combined to obtain the generation probability Pgen. Finally, P_{copy}, P_{vocab} and P_{gen} are combined to get the final word probability distribution. The specific steps are shown in formula (11)˜(13).

$$P_{\text{vocab}} = \mathrm{a}_{\mathrm{t,i}} \tag{11}$$

$$p_{\text{gen}} = \sigma\left(W_y y_t + W_s s_t + W_c c_t + b\right) \tag{12}$$

$$P_{\text{fin}} = p_{\text{gen}} P_{\text{vocab}} + (1 - p_{\text{gen}}) P_{\text{copy}} \tag{13}$$

In addition, during the training process, the model adopts negative log-likelihood loss, and the calculation formula is as in formula (14).

$$\text{loss} = -\frac{1}{l} \sum_{i=1}^{l} \log P(w_t^*) \tag{14}$$

The model's ability to generate questions by leveraging contextual information improves as the overall loss decreases, where the average value is calculated using w_t^* as the target word at time t.

IV. EXPERIMENTS

The SQuAD dataset, consisting of over 100,000 questions sourced from 536 Wikipedia articles and posed by humans, was used to test the enhanced model. It is not public, so this article refers to the practice of Zhao et al. [11] to divide the training, verification, and test sets. The data division is shown in Table 2.

TABLE II: Dataset Statistics

dataset	question num	Avg paragrap len	Avg question len	Avg answerlen len
Train	75722	135	11	3
Valid	10570	138	11	3
Test	11877	130	12	3

This paper uses four baseline systems for comparison:

(1) seq2seq: The conventional approach to neural machine translation involves reversing the input sentence order, optimizing the model parameters using a development dataset, and choosing the model that achieves the lowest perplexity.

(2) seg2seq+Attention: This model generates questions using a global attention-based RNN encoder-decoder framework.

(3) s2s-a-at-mcp-gsa: This model introduces a gate attention mechanism at the encoding end, uses a maximum output pointer network at the decoding end, and can handle long text as input problem generation.

(4) ASs2s [12]: This model proposes an answer-separated sequence-to-sequence model and identifies which latent words in the original text can replace the target answer when generating questions.

TABLE III: Models Question Generation Results

Models	BLEU_4	METEOR	ROUGE_L
seq2seq	4.26%	9.88%	29.75%
seq2seq+attention	12.28%	16.62%	39.75%
s2s-a-at-mcp-gsa	16.38%	20.25%	44.48%
ASs2s	16.20%	19.92%	43.96%
Our Model(no syntax)	**16.51%**	20.19%	**44.53%**
Our Model	**16.59%**	**20.36%**	**44.68%**

For the experimental results, this paper adopts the automatic evaluation index. They commonly used similarity metrics for question generation: BLEU [13], METEOR [14], and ROUGEL [15]. BLUE-N is to calculate the accuracy rate of co-occurrence N-grams (N-grams) relative to the reference sentence. METEOR calculates sentence similarity based on dimensions such as exact match, word stem, synonyms, and free translation. ROUGEL is a metric that quantifies the similarity between generated problems and reference problems by computing an F-measure based on the longest common subsequence. The quality of the generated problem is proportional to the metric's value, which increases as the generated problem approaches the human-generated problem in terms of word combinations. Table 3 displays the results of the experiment.

After conducting experiments, we found that our model has shown significant improvement in multiple metrics for question generation compared to the baseline model. Specifically, the seq2seq+Attention model has outperformed the Seq2Seq model in generating questions. The gated attention mechanism proposed by the s2s-a-at-mcp-gsa model can well represent the input of long text with semantic encoding, which solves the challenge of long text as input in question generation. The quality of the generated questions can be improved by extracting valuable information from the answer, as indicated by various metrics in the ASs2s model. This paper, like s2s-a-at-mcp-gsa, uses long text as input to generate questions. However, because the improved model combines paragraph information and answer sentence information, the indicators generated by the final question are better than s2s-a-at- mcp-gsa. In addition, the input of the decoder adds grammatical information, and the paragraph attention mechanism can capture this information, so the evaluation indicators are better than seq2seq+attention. In order to prove that adding syntactic information helps improve the quality of generated questions, we conducted an ablation experiment. The experimental results show that the improved model with syntactic information is better than the improved model without adding syntactic information.

Model generation question comparison

Example 1:The traditional media day for the game, which usually took place on Tuesday afternoon before the event, was rebranded as the Super Bowl Opening Night and moved to Monday evening.

seq2seq+Attention:What was the media day rebramded as?

Our Model:What was the new designation assigned to the Media Day?

Example 2: If the water level decreases and the temperature of the firebox crown rises considerably, the lead in the system will melt, causing the release of steam as a warning signal to the operators. This will alert them to manually extinguish the fire.

seq2seq+Attention:What is the cause of the <unk>?

Our Model:What happens after the lead melts?

Example 3:In the west, such as England and the USA, the most important food is potatoes.

seq2seq+Attention:What is the most important food?

Our Model:What is the most important food in some western countries?

In this experiment, an artificial scoring system is also employed to evaluate the natural language questions generated by the model. The scoring criteria are divided into four intervals: 3 points: The questions generated by the model are closely

related to the specified answers, and the grammar is coherent. 2 points: The content of the question generally aligns with the specified answer, and the grammar is relatively smooth, with possible minor word errors. 1 point: The content of the question is relevant to the specified answer. 0 points: The generated question is entirely unrelated to the answer. For the experiment, two volunteers with expertise in relevant fields were invited to score 40 questions manually. The results are presented in Table 4.

TABLE IV: Manual scoring results of each model

model	Average score	kappa value
seq2seq+attention	1.58	0.59
Our Model(no syntax)	1.72	0.63
Our Model	1.81	0.67

According to the results in Table 4, the questions raised by the improved model on the SQuAD dataset obtained generally high scores from the two volunteers. This shows that the improved model can generate natural language questions that are more accurate and grammatically fluent.

The box above compares the improved and baseline models' quality for generating natural language questions. There are three groups of samples. Each group of samples contains the original sentence of the Answer and the generated related questions. The underscore in the original sentence is divided into the specified Answer. Example 1, the seq2seq+attention model incorrectly generated "was" and "the" after generating "what" and affected the subsequent generation sequence. This proves that adding grammatical information helps improve the quality of generated questions. As seen from Example 2, adding a copy mechanism and a multi-layer attention mechanism can allow the model to solve the problems of information loss and offset, making the generated questions more accurate and semantically relevant.

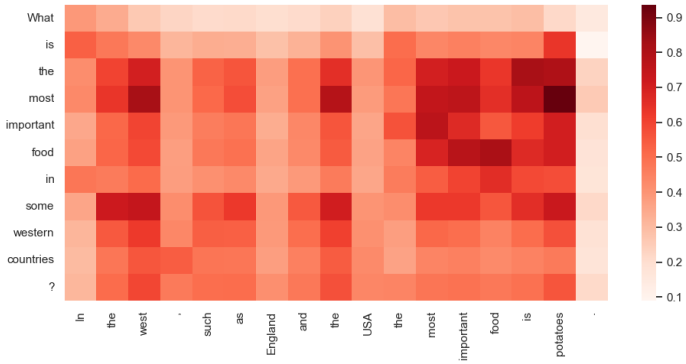

Fig. 2: Heatmap of the attention weight matrix.

Figure 2 shows the global attention weight heat map generated by the improved model, where the highlighted part indicates a high weight, indicating a strong correlation between the corresponding two words. The heatmap presents the correlation between the question generated by the improved model using the sentence in Example 3 and the original answer sentence, highlighting which vocabulary plays a vital role between the question and the answer. The visual presentation of this strong correlation enables us to more accurately understand how the model connects questions and answers during the generation process, further improving the understanding and expression capabilities of the model.

V. Conclusion

In this paper, through splicing and pooling operations, various information such as syntactic information, named entity recognition, and part-of-speech tagging can be fused to form a more comprehensive and richer lexical feature representation, thereby improving the model's ability to understand and express sentences. A multi-level attention mechanism is adopted on the encoding side to capture sequence context information better, improve model expression ability and make up for the lack of local attention, thereby improving the performance and generalization ability of the model in sequence tasks. Experimental results show that exploiting syntactic features and multi-level attention can improve the quality of generated questions. However, the generation question of this model is relatively single, and it cannot generate diverse questions for the same answer. We maintain a strong focus on improving the diversity of question generation [16]. Therefore, in the following stages, we will devote ourselves to in-depth research on increasing the diversity of question generation. We will explore various methods and techniques to enable models to create more varied and diverse generative outcomes.We are greatly inspired by the diversity question generation method proposed by Zhu [17]. This will help improve the model's ability to handle complex problems and meet user needs. Through the efforts of this research direction, we can continuously improve and advance the development of next-generation technology to provide users with more affluent and personalized output.

References

[1] Guillaume Le Berre, Christophe Cerisara, Philippe Langlais, and Guy Lapalme. Unsupervised multiple-choice question generation for out-of-domain q&a fine-tuning. In *60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[2] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[3] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*, 2016.

[4] Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words. *arXiv preprint arXiv:1603.08148*, 2016.

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

[6] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*, 2017.

[7] Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 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)*, pages 569–574, 2018.

[8] Luu Anh Tuan, Darsh Shah, and Regina Barzilay. Capturing greater context for question generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9065–9072, 2020.
[9] Xinya Du and Claire Cardie. Harvesting paragraph-level question-answer pairs from wikipedia. *arXiv preprint arXiv:1805.05942*, 2018.
[10] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
[11] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa 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*, pages 3901–3910, 2018.
[12] Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. Improving neural question generation using answer separation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6602–6609, 2019.
[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
[14] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.
[15] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.
[16] Sen Yang, Qingyu Zhou, Dawei Feng, Yang Liu, Chao Li, Yunbo Cao, and Dongsheng Li. Diversity and consistency: Exploring visual question-answer pair generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1053–1066, 2021.
[17] He Zhu, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Diversity learning based on multi-latent space for medical image visual question generation. *Sensors*, 23(3):1057, 2023.