Chinese Neural Question Generation: Augmenting Knowledge into Multiple Neural Encoders

Ming Liu 1,∗ and Jinxu Zhang 2

1 Faculty of Education, School of Educational Technology, Southwest University, Chongqing 400716, China
2 School of Computer and Information Science, Southwest University, Chongqing 400716, China;
zhangjinxu@email.swu.edu.cn
∗ Correspondence: mingliu@swu.edu.cn

Abstract: Neural question generation (NQG) is the task of automatically generating a question from a given passage and answering it with sequence-to-sequence neural models. Passage compression has been proposed to address the challenge of generating questions from a long passage text by only extracting relevant sentences containing the answer. However, it may not work well if the discarded irrelevant sentences contain the contextual information for the target question. Therefore, this study investigated how to incorporate knowledge triples into the sequence-to-sequence neural model to reduce such contextual information loss and proposed a multi-encoder neural model for Chinese question generation. This approach has been extensively evaluated in a large Chinese question and answer dataset. The study results showed that our approach outperformed the state-of-the-art NQG models by 5.938 points on the BLEU score and 7.120 points on the ROUGE-L score on the average since the proposed model is answer focused, which is helpful to produce an interrogative word matching the answer type. In addition, augmenting the information from the knowledge graph improves the BLEU score by 10.884 points. Finally, we discuss the challenges remaining for Chinese NQG.

Keywords: natural language processing; question generation; deep learning

1. Introduction

Automatic question generation (QG) is an important task in many education applications and tasks. Questions help learners to identify their knowledge deficits and reflect on what they have read [1]. In addition, it is an important component in advanced educational systems [2], such as intelligent tutoring systems and broadly construed dialogue systems. Last, QG can save a large amount of time for test developers with creating assessment instruments, such as reading comprehension tests [3]. QG aims to generate natural language questions based on given contents, including paragraphs [4], sentences, knowledge base triples [5] or images [6]. In this study, we focus on the QG task based on the given passage and target answer. The answer is a text span of the passage text, and the system-generated question should be similar to a human question.

Previous works for QG in both Chinese and English have mainly employed rule based approaches consisting of a set of rules that encode a set of heuristics that basically transform a declarative sentence into an interrogative one [3,4,7]. Rule-based approaches are easy to implement purely based on syntactic information (e.g., subject, main verb, object), semantic information (e.g., named entities), and human-designed transformation rules (e.g., wh-movement). However, questions generated by this approach are generally stereotypical. In addition, the quality of these questions is low since the question generation is a pipeline process that includes many imperfect stages that introduce errors, such as syntactic and semantic parsing errors, and mismatched QG rules. Errors introduced in early stages of the pipeline propagate all the way to the output multiplying its negative effects on later stages.
of processing and ultimately the output. A different architecture, such as the one proposed here, would make many of these errors avoidable and their multiplying effect eliminated or substantially reduced.

In recent years, with the availability of “big data”, such as large question banks (SQuAD; [8]), and effective machine learning algorithms, e.g., deep neural networks [9], data driven approaches based on new data processing architectures have attracted a great deal of interest from researchers working on natural language processing tasks, such as neural text summarization [10] and neural machine translation [11], mainly because these approaches do not require human defined rules and have good generalization power. Some researchers attempted to adopt this approach in question generation [12–14]. Serban et al. [13] presented a recurrent neural network to generate factoid English questions from structured data, which is not common in text-to-text question generation tasks, where the source is an unstructured text, such as a sentence or paragraph. Du et al. [12] proposed a preliminary study on neural English question generation from text by using an encoder-decoder neural network model that takes an answer and its source text as input and outputs a relevant question. Zhou et al. [14] further improved the model with rich features about the answer (e.g., answer position and lexical features) to generate answer focused questions, and incorporate a copy mechanism, dealing with unknown words, that allows the copy of the source text when generating questions. This model has reached one of the best results on the Stanford Question Answering Dataset (SQuAD), which contains 100,100+ question answer pairs based on the Wikipedia articles. However, the error analysis results showed that most ill generated questions (37.04% errors) had incorrect question words or interrogative words although the answer position feature has been incorporated.

Another problem is that these erroneous questions (20.37% errors) include wrong parts of the source text, which are irrelevant to the answer. A major cause for these problems is the relatively large size of the source text, which may cause the current neural encoder–decoder model to not pay attention to the answer and relevant context words. To address the complexity of the long passage text, a traditional approach includes a passage compression step in question generation by extracting only relevant sentences containing the answer [15]. Thus, the question is generated based on the relevant sentences. Another approach is based on the recent development of attention-based neural models, such as self-attention [16] or multi-head self-attention mechanism [11] in encoding passages. These models may automatically pay higher attention to the target answer and its surrounding words. Nevertheless, these approaches may not work well when the contextual words, where the question generates, do not contain the answer or pronoun. This often happens in Chinese since the target answer or pronoun is implicitly described or missing in Chinese, also called zero-pronoun phenomenon [17]. In the following Example Passage1, the italicized he and He in English are omitted in Chinese.

Table 1 gives an example of a question generated by using Zhao et al.’s model [16] from the shown passage containing four sentences. In this case, the model generates an unacceptable question with a wrong question word,” What did Xu Beihong teach in Peiping Peiping?”. It is because the passage is too long and difficult for the model to pay attention to the right context words (e.g., Running Horse) and use of the right question word “who”. In addition, it suffers from the repetition errors since the copy mechanism tends to copy high frequency words (e.g., 北平 Peiping) in the passage.
1. To address Chinese QG from the long source text issue, we introduced a
passage compression step from previous question generation research [15] and introduced a
novel approach to addressing the problem of contextual information loss because of passage
compression by augmenting the neural question generation model with Chinese Wikipedia
knowledge triples. To better incorporate different information sources (e.g., compressed
passage, knowledge triples and answers), a multi-encoder sequence-to-sequence neural
model was proposed with an answer focused attention mechanism. This
model was evaluated by automatic machine evaluation and human judgements based on
a large Chinese question and answer dataset. Finally, we have discussed the challenges
for Chinese neural question generation. Specifically, the contribution of this paper can be
summarized as follows:

1. To address Chinese QG from the long source text issue, we introduced a
passage compression step and incorporated Chinese knowledge graphs to compensate for
the information loss of the passage compression. This approach was inspired by the
extracting relevant sentence step in question generation [15], whose purpose was to
minimize question generation complexity from long source texts. However, to reduce the
loss of context information from the answer, we incorporated Chinese Wikipedia
knowledge to the neural model by extracting and ranking triples based on their text
similarity to irrelevant sentences. We investigated the impact of the number of triples
used on the system performance.

2. To improve the answer-focused attention mechanism, we proposed a multi-encoder
sequence-to-sequence neural model with a double attention mechanism, where compressed
text, triples and answer information have been encoded separately. The
global attention mechanism pays attention to each encoder, while the local attention
mechanism of each encoder pays attention to each word. Therefore, the answer and its
corresponding encoder will be focused. We investigated the impact of different encod-
ing schemas, ranging from one to three encoders, on the system performance. More-
over, inspired by recent success of the Bidirectional Encoder Representations from
Transformers (BERT) model in NLP tasks [19,20], we used BERT to encode the text,
triples and answer information due to its rich contextual information representation.

3. We performed comprehensive human evaluation and error analysis. Besides the ques-
tion acceptability, we introduced a new evaluation metric, called the same question
type ratio (SQTR), which measures the ratio of the generated questions that have the same question types as the reference question. Human experts manually analyzed 800 questions from each neural model and annotated each question acceptability and question type. Moreover, an error analysis of the Chinese questions was also performed. This kind of analysis provides valuable insights for future work with respect to how to improve Chinese question generation performance.

2. Related Work

2.1. Paragraph-Level Question Generation

Utilizing paragraph-level contexts around the input text is an important factor to generate good questions. Du et al. [12] observed that 20% of questions in the SQuAD dataset require paragraph-level information to be answered. However, as the input texts get longer, it becomes more challenging for sequence-to-sequence neural models to learn to “copy” relevant context and avoid irrelevant information. To overcome the challenges of processing long sequences of text inputs, Zhao et al. [16] introduced a gated self-attention encoder with a maxout pointer mechanism. The gated self-attention encoder first takes an encoded passage-answer representation as input and performs matching against itself to compute a self-matching representation. Then, the encoder combines the input with the self-matching representation using a feature fusion gate. The decoder used a maxout pointer mechanism [21] to overcome the repetition problem (the repeated occurrence of words in the input text tends to cause repetitions in output sequence). The study results showed that the gated self-attention mechanism helped the decoder focus on the context words surrounding the answer, and thus reached good performance in question generation. Nonetheless, we argue, if the context words used for human question generation are far from the answer, the model may fail to generate questions “as natural” as humans generate. To capture richer contextual representation of passage, Chan and Fan [19] used the Bidirectional Encoder Representations from Transformers (BERT) to encode the passage. The decoder only contains a fully connected layer connecting from the output of the last token hidden state in the input sequence. They found that this BERT passage encoder outperformed Zhao’s question generation model on English SQuAD dataset.

Instead of using the whole paragraph as context information for question generation, Du and Cardie [15] considered only relevant sentences, i.e., sentences containing the answer. In addition, co-reference information in the sentences served as an extra feature into the encoder. The co-reference information is obtained by a gate mechanism [22]. The study results showed that the co-reference representation improved their question generation system significantly. However, in Chinese the anaphora is often missing, called zero pronouns, which brings challenges with respect to explicitly capturing their semantic similarities with antecedents [17]. This will cause a major challenge with obtaining correct co-reference information in Chinese paragraphs. Inspired by Du and Cardie’s work, we introduce a passage compression step, which only selects relevant sentences containing the answer. However, this process has its own challenge: it may lose useful information about the answer. Therefore, we added information from a knowledge graph to compensate for the loss of irrelevant sentences. A knowledge graph or base is a structured database, which contains entities and relationships between those entities usually in the form of a triple (entity1, relation, entity2). In recent years, several large Chinese encyclopedia knowledge graphs were made publicly available on cloud, such as CN-DBpedia2 [23]. For example, CN-DBpedia2 [23] contains around 16,024,656 entities and 228,499,155 relation types. It was developed based on Baidu Baike [24], Hudong Baike [25], Chinese Wikipedia [26] and other encyclopedia websites. Recently, deep neural networks have been used to generate factual questions based on triples in the CN-DBpedia2 knowledge graph [27]. In the current study, we extracted and ranked the triples related to the answer entity and which are semantically similar to the filtered sentences. Thus, the filtered information about the answer is largely retained and to some degree expanded.
2.2. Answer-Focus Question Generation

Du et al. [12] pioneered the first NQG model using an attention Seq2Seq model [11], which inputs a sentence into an RNN-based encoder and produces a question about the sentence using a decoder. The attention mechanism is used to assist the decoder pay attention to the most relevant parts of the input sentence while producing a question. However, this base model does not encode the target answer, thus may not be able to learn which information to emphasize for question generation.

Several NQG models have investigated encoding the target answer information by either treating the answer’s position as an extra input feature [14,16] or by encoding the answer with a separate “RNN” [28,29]. The first type of methods encodes each input word vector with an extra answer feature indicating whether this word is within the answer span. Zhao et al. [16] used the BIO (denotes the beginning, inside and outside of an answer) tagging scheme to implement this feature, while Harrison and Walker [30] directly adopted a binary indicator. Sun et al. [31] argues that the context words closer to the answer also require more attention from the model. Thus, they encoded position embeddings into the computation of the attention distribution, where the position distance between the word and answer indicated the level of attention. The study results showed that the model reached an extra BLEU4 percentage of 0.89 on the SQuAD. However, adding the answer feature may tend to generate questions that include answer words [29]. To address this issue, Kim et al. [27] proposed a separate RNN encoder for the answer such that the answer is replaced by a special token for passage encoding. Inspired by Kim et al.’s work, in our proposed multi-encoder sequence-to-sequence question generation framework, the answer information was encoded separately, and we investigated the impact of the encoding schema on the accuracy of question types and overall quality of generated questions.

2.3. Deep Transfer Learning

Over the last 3 years, the emergence of transfer learning methods and models in the field of natural language processing greatly improved upon the state-of-the-art on a wide range of NLP tasks, including text generation [32]. Transfer learning contains two phases: (1) A pretraining phase where contextual representations are learned on a source task; and (2) An adaptation phase in which the learned knowledge is applied to a target task or domain. The unsupervised pre-training models, such as BERT [33], GPT [34], BART [35] and T5 [36], have accomplished great success in NLP because unlabeled text data is available at large on the internet.

Devlin et al. [31] designed a bidirectional encoder representation (BERT) model pretrained on large corpus of text using a masked language model to capture better contextual representation of text. Based on their work, Dong et al. [37] and Bao et al. [38] proposed a unified pretrained language model which modified BERT for a text to text generation task using self-attention masks. Compared with BERT, the GPT model is auto-regressive in nature and contains stacked decoder blocks from the transformer architecture. GPT is suitable for text to text generation tasks.

Colin et al. [36] introduced a unified framework based on the BERT-like encoder and GPT-like decoder Transformer architecture (T5) that converts all text-based language problems into a text-to-text format and achieved state-of-the-art results on many NLP tasks by combining a new “Colossal Clean Crawled Corpus” with the insights from the exploration on pre-training objectives and architectures. Similarly, Lewis et al. [39] proposed a transformer-based neural machine translation architecture, called BART, and investigated several denoising schemes used for pretraining the model. They found that BART was particularly effective when finetuned for language generation with text infilling schemes. However, many of these language models were pretrained solely on English-language text. More recent transfer learning approaches focused on multilingual models that have been pre-trained on a mixture of many languages, such as mT5 [40] and mBART [41]. In this study, we have compared BERT and mT5 with our proposed model.
3. Our Neural Question Generation Approach

3.1. Passage Compression Step Using the Knowledge Graph

Passage Compression: the objective of this step is to compress the source paragraph without losing important context information about the answer for question generation. Figure 1 shows the three main steps for paragraph/passage compression. Sentence splitting, and pronoun resolution are carried out at the pre-processing step. Step 1 involves breaking the passage into relevant sentences and irrelevant sentences, which is similar to Du and Cardie’s work [15]. A relevant sentence contains the answer directly, while irrelevant sentences do not contain the answer. A simple pronoun resolution step has been performed by checking if the sentence explicitly contains a pronoun and replacing each with the nearest named entity.

![Figure 1. Passage Compression and Knowledge Argumentation Steps.](image-url)

In order to compensate for contextual information loss resulting from discarding the irrelevant sentences, steps 2 and 3 link the answer to a Chinese Wikipedia knowledge graph, CN-DBpedia2 [18]. Triples are extracted from the graph and ranked based on the similarity between triples and irrelevant sentences. We used the text similarity algorithm proposed by Huang et al. [42], which first extracts the key terms from the irrelevant sentences to represent the overall text using TF-IDF scores, and then proposed the term similarity weight tree based on the Chinese WordNet semantic distance to compute overall similarity between the key terms and a triple. This algorithm is simple, but very suitable for computing short text similarity. As a result, the passage compression leads to two types of data, the compressed passage (consisting of relevant sentences) and the top ranked similar triples (representing the semantic information about irrelevant sentences). Table 2 shows an example of the passage compression results from the original passage shown in Table 1. In example passage 1, only the first sentence explicitly contains the answer and extracted as a relevant sentence. The other three sentences are considered irrelevant even though they contain zero-pronouns. Based on the computed similarity scores between the irrelevant sentences and knowledge graph triples, the top three ranked triples were selected. The answer appears in every triple and does not provide much contextual information, so the answer word in each triple is removed before encoding them into the model. We discuss the impact of the number of triples incorporated into the neural model on system performance in Section 4.3.1.
Table 2. An example of compressed passage shown in Table 1 with triples extracted from CN-
DBpedia linking to Xu Beihong (Xu Beihong).

Irrelevant Sentences: 赣留学法国学西画，归国后长期从事美术教育。先后任教于北平国立中央大学艺术系、北平大学艺术学院和北平艺专。擅长人物、走兽、花鸟，主张现实主义，其代表作有《群马》、《奔马图》、《愚公移山图》。晚年去世后，1200余件作品全部捐献给国家。He studied in France and studied Western painting, and then engaged in art education for a long time in China and taught in the Department of Art of National Central University in Peiping, the School of Art of Peiping University, and the Peiping Art College. He was good at characters, beasts, flowers and birds, and advocates realism and his representative works included “Gunma”, “Running Horse”, and “Yugong Yishan”. After he died, his wife donated more than 1200 pieces of his works to the country.

Relevant Sentences: 徐悲鸿, 代表作品, 《奔马图》 Xu Beihong, formerly known as Xu Shoukang, was a modern Chinese painter and art educator. Top 3 ranked triples from knowledge graph:
[Xu Beihong, Art work, “Running Horse”]
[Xu Beihong, Art work, “Gunma”]
[Xu Beihong, Art work, “Yugong Yishan”]

3.2. Multiple Encoders Sequence-To-Sequence Model with Double Attention Mechanism

Figure 2 shows the architecture of a multi-encoder sequence-to-sequence neural model in question generation, called Multi-encQG. Each encoder Bidirectional Encoder Representations from transformers iterates over the sequence produced by the respective data source.

3.2.1. Multi-Encoders

As suggested by Chan and Fan [19], the BERT encoder captures rich contextual information of passage, which has the benefit for question generation. To ensure the model does not lose focus on the answer, we adapted the BERT model for encoding compressed passage, triples and answers separately as follows. First, for a given compressed passage $P = [p_1, \cdots, p_m]$, a list of triples $T = [t_1, \cdots, t_q]$, and $A = [a_1, \cdots, a_l]$, the input sequences are denoted as $\text{passage} = ([CLS], P, [SEP])$, $\text{triples} = ([CLS], T, [SEP])$, $\text{answer} = ([CLS], A, [SEP])$. Each triple is encoded as a sequence of words. The answer appears in every triple and does not provide much contextual information, so the answer word in each triple is removed before encoding them into the model. Second, each input sequence is represented by the BERT embedding layers. Last, we chose the hidden state of the last layer in the BERT
model corresponding to each input token. Thus, the contextual BERT embeddings for the compressed passage, triples and answer are denoted as follows:

\[ h_{\text{passage}} = [h_{\text{cls}}, h_1, h_2, \ldots, h_m] \] (1)

\[ h_{\text{triples}} = [h_{\text{cls}}, h_1, h_2, \ldots, h_n] \] (2)

\[ h_{\text{answer}} = [h_{\text{cls}}, h_1, h_2, \ldots, h_l] \] (3)

where \( h_{\text{cls}} \) relates to the hidden state of <CLS> token and \( h_i \) represents the \( i \)-th token of the input sequence.

### 3.2.2. Double Attention-Based Decoder

We extended previous attention-based question decoder approaches [14,31] by employing a combination of global and local attention mechanisms to decode the passage, triples and answer information to generate questions. The question words are encoded by using the BERT model since this pre-trained transformer can capture rich contextual embedding of decoded question words. Inspired by Chan and Fan’s work [19] on using the BERT model to decode questions by adding a <MASK> token to capture the previous decoded results, we added a <MASK> token to the end of decoded question words in our model. The question words are denoted as question = ([CLS], [START], Q, [MASK]), where \( Q \) is \([q_{\text{cls}}, q_{\text{start}}, q_1 \cdots q_{\text{last}}, q_{\text{mask}}]\). \( q_{\text{last}} \) is the last decoded question token. The hidden state of the last BERT layer for each corresponding question word is denoted as follows:

\[ s_{\text{question}} = [s_{\text{cls}}, s_{\text{start}}, s_1, \ldots, s_{\text{last}}, s_{\text{mask}}] \] (4)

where \( s_{\text{mask}} \) is the hidden state of the current word decoding and \( s_{\text{last}} \) refers to the hidden states of the last decoded question token.

The proposed attention mechanism emphasizes the global attention distribution over encoders and the local attention distribution of each encoder over input characters. The global attention weights indicate how the model spreads out the amount it cares about to different encoders. At time step \( t \), the global attention weights for an \( i \)-th encoder \( a_{t, i}^{\text{global}} \) and context vector \( c_t \) are calculated as follow:

\[ e_{t,i} = v^T \tanh(W_h \text{Encoder}_i(a_{\text{local},h}) + W_s s_{\text{last}} + b_{\text{attn}}) i = 1, 2, 3 \] (5)

\[ a_{t,i}^{\text{global}} = \text{softmax}(e_{t,i}) \] (6)

\[ c_t = \sum_{i=1}^{n} a_{t,i}^{\text{global}} \text{Encoder}_i(a_{\text{local},h}) \] (7)

where \( W_h, W_s + b_{\text{attn}} \) are all trainable parameters and \( s_{\text{last}} \) is the decoded state of the previous question token. \( \text{Encoder}_i(a_{\text{local},h}) \) is the local context for the \( i \)-th encoder which is calculated:

\[ \text{Encoder}_i = \sum_{c=1}^{n} a_{t,c}^{\text{local}} h_c \] (8)

where \( n \) is the number of input tokens in the \( i \)-th encoder and \( a_{t,c}^{\text{local}} \) represent the local attention weights for each token \( c \). \( h_c \) is the token \( c \)'s hidden state shown in the encoder section. \( a_{\text{local}} \) is computed based on \( h_c \) and \( S_t \) through the concatenate attention mechanism [11].

Based on the previous decoded question token embedding \( w_{\text{last}} \), the current decoder state \( s_{\text{mask}} \) and the current context vector \( c_t \), a readout state \( f_t \) is computed by an MLP maxout layer with dropouts. Finally, the readout state is passed to a linear layer and a softmax layer to predict the probabilities of the next word over the decoding vocabulary.

\[ f_t = w_a w_{\text{last}} + w_b c_t + w_s s_{\text{mask}} \] (9)
During the training phase, the objective is to minimize the difference between the generated question tokens and the reference in the training set. The difference or loss is quantified by using the following negative log likelihood for each question:

\[ L(\theta) = - \log P(q_t|C, A, K, \{q_i\}_{i=0}^{t-1}, \theta) \]  

where \( C \) is the compressed passage tokens, \( A \) is the answer tokens, \( K \) is the knowledge triples, \( |q_t| \) is the number of decoded question tokens.

4. Experiments

4.1. WebQA Dataset Description

WebQA is a large scaled Chinese question and answer dataset developed by Baidu AI researchers [39] and consists of tuples of (question, evidence, answer) which is like the example in Table 1. The questions and answers are mainly collected from a large online Chinese QA community website called Baidu Zhidao and those questions are asked by real-world users in daily life. All the evidence is retrieved from the internet by using Baidu search engine with questions as queries and are annotated by humans as positive or negative evidence. Positive evidence means this passage with around five sentences contains the answer. For this experiment, we have used the 37,354 tuples (question, positive evidence, answer) with these answers which can link to the Chinese knowledge graph. Within these tuples, each pair of question-answer is related to only one passage or positive evidence. No more than two questions are related to one answer. This dataset was divided into a training set containing 29,884, a validation set 3735 and a testing set containing 3735.

Table 3 shows that the average number of words per passage has been decreased by around 49% in the training set and 59% in the testing set after performing the passage compression step described in Section 3.1.

| Dataset | Avg Number of Words Per Passage | Avg Number of Words Per Passage after Compression | Number of Question–Answer Pairs |
|---------|---------------------------------|-----------------------------------------------|--------------------------------|
| WebQA   | 75.44                           | 38.92                                         | 29,884                         |
| Validation | 72.25                          | 31.55                                         | 3735                           |
| Test    | 71.89                           | 29.75                                         | 3735                           |

We implemented our multi-encoders sequence-to-sequence neural model in TensorFlow 1.4 [34], which was trained on a single NVIDIA GeForce GTX 1080 Ti Graphics Card. The compressed passage, triple, answer and question inputs are first tokenized, and then encoded by using the officially provided pre-trained BERT\textsubscript{base} model which contains 12 layers, 768 hidden dimensions, and 12 attention heads [31]. The encoder and decoder used the same BERT vocabulary of 21,128 tokens since they are in the same language. The dropout [43] probability is set to 0.3 between transformer layers. The cut-off values for the length of the compressed passage, triples, question and answer are 80, 30, 20 and 10, respectively in the WebQA dataset. All model parameters are optimized using Adam optimizer [44] with \( \beta_1 = 0.9, \beta_2 = 0.999 \). The initial learning rate of \( 11 \times 10^{-4} \) with linear warm-up over the first 10,000 steps and linear decay. The parameters of models are learned to minimize the cross-entropy loss computed using the predicted tokens and the reference tokens. The mini-batch size for each update is set as 32 and our models are trained up to 8 epochs. A final merge operation was performed on the output of the decoded question.
sub words defined in the BERT vocabulary. We use the Beam Search strategy for sequence decoding. The beam size is set to 3.

4.2. Comparative Neural Question Generation Models

We have explored the following models to compare with prior work from question generation in the literature:

1. Trans-Rules: the rule-based Chinese question generation system [7], which transforms a declarative sentence into an interrogative question by using predefined rules. We modified the code so that it can generate questions based on a given passage and the answer.
2. s2s+ Att: baseline encoder-decoder based seq2seq network with attention mechanism [12].
3. NQG++: the state-of-the-art neural question generation system extends s2s+Att by adding answer position and linguistic features to the embedding layer and a copy mechanism [11]. The Chinese word embeddings are pre-trained on large-scale Chinese Wikipedia and Baidu dataset [45].
4. NQG++MP: extends NQG++ model by adding a maxout pointer mechanism with a gated self-attention encoder [16].
5. NQG++PC: extends NQG++ by adding the passage compression step described in Section 3.1.
6. BERT-SQG: uses the pre-trained BERT model to encode passage and previous decoded question words and a single fully-connected layer to decode question words based on the hidden state of last token on the last layer in the BERT model [19].
7. mT5: uses the pre-trained mT5-base model, where the encoder handles the input passage and answer, and the decoder generates question words [41]. Bert4keras library is used for the model implementation [46].
8. Single-encQG: our proposed model with only one BERT encoder compressed passage, answer and triples, and deploys a double attention-based decoder combining with the question word BERT representation.
9. Multi-encQG +TA/P: our proposed model with two encoders: one BERT encoder for passages and another BERT encoder for triples from the knowledge graph and answer embeddings.
10. Multi-encQG +PA/T: our proposed model with two encoders: one BERT encoder for passages and answer, and another BERT encoder for triples from the knowledge graph.
11. Multi-encQG +PT/A: our proposed model with two encoders: one BERT encoder for passages and triples, and another BERT encoder for the answer.
12. Multi-encQG ++: our proposed model with three encoders for passages, triples and answers, respectively.
13. Multi-encQG-triples: ablation test; the triple BERT encoder is removed from Multi-encQG++.
14. Multi-encQG-answer: ablation test; the triple Answer BERT encoder is removed from Multi-encQG++.

4.3. Machine Evaluation and Results

We have used the following two automatic evaluation metrics commonly used in neural question generation [39], which originally designed for machine translation and summarization.

BLEU [34] metric scores a question on a scale of 0 to 100, to measure the adequacy and fluence of the system output. The closer to 100 the BLEU score, the more overlap there is with the corresponding human question, which is considered a gold standard. BLEU-1, BLEU-2, BLEU-3 and BLEU-4 use 1-g to 4-g for this overlapping calculation, respectively.

ROUGE-L [33] measures by comparing a system generated question against a human question. In this study, we used ROUGE-L, which measures the longest marching sequence of words using the longest common subsequence, which accounts for matching sentence level word order.
4.3.1. Comparison Results with Different Question Generation Models

We compare our model with approaches that have open source codes. Table 4 shows that our proposed Multi-encQG++ model outperformed the BERT-SQR and NQG++MP, which were the state-of-the-art QG models, by at least 7.629 average BLEU score points and by 9.81 ROUGE-L score points in the WebQA dataset. This is mainly due to the passage compression step which has reduced the complexity of the whole QG process. Another reason is the multi-encoder sequence-to-sequence neural model with a double attention mechanism which allows the decoder to focus on the answer and generate correct question words. We will examine the correctness of question words in the Human Evaluation Section.

Table 4. Machine evaluation results of different question generation models on the Baidu Q and A dataset. Top 3 ranked triples were incorporated into Multi-encQG models.

| Dataset  | Model            | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L |
|----------|------------------|--------|--------|--------|--------|---------|
| WebQA    | Trans-Rules      | 23.21  | 12.00  | 7.24   | 4.51   | 18.02   |
|          | s2s+Att          | 23.61  | 12.96  | 7.99   | 5.05   | 19.15   |
|          | NQG++            | 28.06  | 15.66  | 9.82   | 6.45   | 23.97   |
|          | NQG++PC          | 24.65  | 13.04  | 8.09   | 6.06   | 21.16   |
|          | NQG++MP          | 33.32  | 18.78  | 12.65  | 8.12   | 29.05   |
|          | BERT-SQR         | 34.14  | 20.13  | 13.45  | 8.98   | 30.47   |
|          | mT5              | 36.52  | 21.45  | 15.24  | 9.88   | 32.58   |
|          | Single-encQG     | 30.12  | 15.38  | 9.35   | 4.92   | 25.72   |
|          | Multi-encQG+TA/P | 32.52  | 17.75  | 12.05  | 8.08   | 28.85   |
|          | Multi-encQG+PA/T | 31.75  | 16.19  | 11.85  | 7.78   | 27.50   |
|          | Multi-encQG+PT/A | 34.10  | 19.76  | 13.21  | 8.37   | 30.13   |
|          | Multi-encQG++    | 42.19  | 21.58  | 16.15  | 9.51   | 35.87   |
|          | Multi-encQG-Triple | 26.85  | 14.58  | 8.58   | 6.94   | 23.76   |
|          | Multi-encQG-answer | 23.98  | 13.21  | 8.34   | 5.28   | 20.65   |

The results also showed that BERT-SQR outscored NQG++MP and other question generation models in the dataset, which was consistent with the finding in Chan and Fan’s study [19], indicating the power of BERT contextual input passage representation for question generation. Moreover, mT5 slightly outperformed BERT-SQR, which demonstrated that the encoder-decoder architecture is more suitable than the single BERT-based language model for text generation task [43]. We also note that the NQG++ model outperforms the Trans-Rules and s2s+Att by a large margin in terms of BLEU and ROUGE-L scores. This result confirms the importance of the answer features encoded for question generation proposed by Zhou et al. [15]. Interestingly, when the passage compression step is added to the NQG++ model, the resulting NQG++PC model performance decreases because of the loss of contextual information.

Lastly, we investigated the impact of encoding schemes in Multi-encQG++ and observed that the Multi-encQG++ PT/A model with a separate answer encoder schema outperformed the Multi-encQG++ with one and the other two encoders since the Multi-encQG++ PT/A pays more attention to the answer when decoding. These results indicated the importance of encoding answer information in a separate encoder.

It has been observed that the BLEU-4 and BLEU-3 scores were relatively lower than BLEU-1 and BLEU-2 due to the small number of overlapping long text between human question and system question. In fact, the questions generated by neural models tend to use synonyms for the human question where the BLEU evaluation metrics cannot detect it (see an example in Table 5). In addition, human questions have perfect punctuation, such as “,” “《” and “》”, whereas the system question often does not contain this punctuation. This would cause different tokenization results of some proper terms, such as “奔马图” (running horse), when computing the BLEU scores. A number of English neural question generation studies reported similar findings [19,31].
Table 5. Questions generated from the example in Table 1 based on different models.

| Model          | Question                                                                 |
|---------------|--------------------------------------------------------------------------|
| Human Reference | “奔马图”是谁画的？Who painted “Running Horse”?                              |
| Q1:谁是中国现代画家、美术教育家?                          |
| Trans-Rules   | Who was a modern Chinese painter and art educator?                        |
| Q2:艺术作品是什么?                                  |
| s2s+Att       | What was the art work?                                                  |
| Error: Vague   | Q3: 徐悲鸿北平任教什么？                                                  |
| NQG++         | What did Xu Beihong teach in Peiping?                                    |
| Q4:谁的原名为徐寿康？                               |
| Error: Grammatical and repetition errors               |
| NQG++MP       | Who was known as Xu Shoukang?                                            |
| Q5:徐寿康是哪位？                                      |
| Error: coping wrong context words                      |
| BERT-SQG      | Who is Xu Shoukang?                                                     |
| Q6:徐悲鸿原名是什么？                                 |
| Error: copying wrong context words                     |
| Multi-encQG   | What was the former name of Xu Beihong?                                  |
| Q7:奔马图是谁的作品？                                |
| Error: incorrect answer                                |
| Multi-encQG++ | Whose artwork was Running Horse?                                         |
| Acceptable    |                                                                           |

In this case, Q1 generated by the Trans-Rules is semantically and grammatically correct but quite different from the human generated question because it does not take into consideration the full paragraph context, which leads to a less naturally-sounding question. Q2, generated by the neural baseline model, is unclear and unrelated to the answer because this model does not encode the answer information and does not know what context words to pay attention to. Q3, generated by the NQG++ model, has grammatical errors because the passage is too long and difficult for the model to pay attention to the right context words. In addition, it suffers from the repetition errors since the copy mechanism tends to copy high frequency words (e.g., 北平 Peiping) in the passage. NQG++MP claimed to solve the repetition-induced errors with the max pointer mechanism [16] and focused on the context words surrounding the answer using a self-attention mechanism. We observe that question Q4, which is generated by the NQG++MP model, is grammatically and semantically correct, has a correct question word, but is different from the human-generated question since the model focused on copying the wrong context words, 原名徐寿康 (formerly known as Xu Shoukang), from the relevant sentence. The zero-pronoun in Chinese may cause the self-attention mechanism to focus on the relevant sentences explicitly containing the answer. This also causes a problem for the BERT-SQG model with multi-head self-attention mechanism. Lastly, Q7 generated by the Multi-encQG++ has a correct question word and copied the right context words, 代表作品奔马图 (representative work, Running Horse). Q6, generated by the Multi-encQG model with a single encoder, has an incorrect answer compared to the human-generated question since the model pays more attention to 徐寿康 (Xu Shoukang) in the compressed text. To better understand how double-attention works for the answer-focused Multi-encQG++ model, we visualized the global and local attention distribution at each time step of decoding question Q7 and compared the attention distribution for Multi-encQG with a single encoder for decoding question Q6.

4.3.2. Ablation Studies

We have investigated the effect of the two encoded information, including answer and triples, on the Multi-encQG++ model performance. After we removed the answer information, the performance of the Multi-encQG++ decreased dramatically by 14.105 average BLEU score in WebQA dataset (see the Multi-encQG-answer in Table 4). This result highlighted the importance of answer information encoded for neural question generation, which had been also found in previous neural English question generation studies [14,31]. Although triple information extracted from the knowledge map is less important than the answer information for question generation, after removing the triple information, the MultiencQG++ performance dropped by 12.57 BLEU scores in WebQA dataset (see the Multi-encQG-triple in Table 4). In other words, augmenting the knowledge triples into the Multi-encQG++ can increase the performance by 10.884 BLEU scores on average.

In addition, we investigated the optimal number of triples used for encoding the Multi-encQG++ model since the more triples were used the more complexity was added. Figure 3 shows that the Multi-encQG++ model performance decreases when encoding more than 3 triples, which shows that when more triples are being used, less relevant
information is being added; thus, the more challenging the model is to learn. We also calculated the number of words in knowledge triples appeared in a reference question.

**Figure 3.** The impact of the number of ranked triples encoded on the Multi-encQG++ model performance in WebQA dataset.

4.4. Case Study

Table 5 provides six example questions generated from the example passage 1 by six question generation models. Compared to the neural models, the rule-based QG approach can easily transform a simple sentence into a question by detecting the named entity “徐悲鸿 Xu Beihong”.

Figure 4 shows that the Multi-encQG model pays more attention to the relevant sentence, particularly 徐寿康 (Xu Shoukang) when generating an incorrect question word 什么 (what), which leads to Q6 having the incorrect question type and answer compared to the human-generated question, which is produced based on the irrelevant sentences. On the other hand, the Multi-encQG++ more pays more attention globally to the answer encoder and the knowledge graph information encoder shown in Figure 5. As a result, the overall double attention mechanism focuses on the context words surrounding “Running Horse” and answer “Xu Beihong”, shown in Figure 6, generating question Q7, which uses a correct question word and is like the human-generated question. Although the Multi-encQG++ model tends to generate more accurate question words, the model does not always copy the right context words from the selected triples or compressed passage. We will discuss this issue later in the error analysis section.

4.5. Human Evaluation and Results

Compared to machine evaluation of question generation, human evaluation provides more reliable assessment of the quality of questions but is more labor intensive. In this study, we evaluated s2s+Att, NQG++, NQG++MP, BERT-SQG, Multi-encQG and Multi-encQG++ models with human judges. Three Chinese linguistic majors randomly selected 800 passages from the test set in the dataset and labelled 800 questions generated by each model from the dataset. Each question was judged as acceptable or unacceptable following a typical judgment protocol used in Chinese question generation [7], which is based on the following major criteria: (1) grammaticality correctness, (2) semantic correctness and (3) correct match of the reference answer and question type. The inter-rater reliability of the question acceptance was measured by using Fleiss [47], which indicated moderate or higher agreements (scores ranged from 0.42 to 0.67) between human annotators for the set of questions generated by each model.
can easily transform a simple sentence into a question by detecting the named entity "徐悲鸿" Xu Beihong.

Figure 4 shows that the Multi-encQG model pays more attention to the relevant sentence, particularly "徐寿康" (Xu Shoukang) when generating an incorrect question word "什么" (what), which leads to Q6 having the incorrect question type and answer compared to the human-generated question, which is produced based on the irrelevant sentences. On the other hand, the Multi-encQG++ more pays more attention globally to the answer encoder and the knowledge graph information encoder shown in Figure 5. As a result, the overall double attention mechanism focuses on the context words surrounding "Running Horse" and answer "Xu Beihong", shown in Figure 6, generating question Q7, which uses a correct question word and is like the human-generated question. Although the Multi-encQG++ model tends to generate more accurate question words, the model does not always copy the right context words from the selected triples or compressed passage. We will discuss this issue later in the error analysis section.

Figure 4. Attention distribution when Multi-encQG generated Q6.

Figure 5. Global attention distribution when NQGKG++ generated question Q6.
This could be explained by the larger number of what and who questions being present in the training set. Models reached better recalls for what and who questions than where and which questions. The baseline s2s+Att model reached the lowest attention mechanism. Compared to the Single-encQG, the Multi-encQG++ with the double attention mechanism increased the acceptance score by 8% because the Multi-encQG++ model is more answer focused and more likely to use the right question words. In fact, it has been found that these models reached better recalls for what and who questions than where and which questions. This could be explained by the larger number of what and who questions being present in the training set.

Table 6 shows that the Multi-encQG++ model reached the best acceptance among six question generation models and slightly outperformed mT5 by 3% in the dataset mainly because the complexity of the QG process is reduced using the passage compression step and the answer-focused multi-encoder sequence-to-sequence neural model with double attention mechanism. Compared to the Single-encQG, the Multi-encQG++ with the double attention mechanism increased the acceptance score by 8% because the Multi-encQG++ model is more answer focused and more likely to use the right question words. In fact, Table 7 shows that the Multi-encQG++ model reached the best SQTR across four main question types, indicating this model can pay better attention to the answer and decode more accurately the question words. The baseline s2s+Att model reached the lowest acceptance since it lacks focus on the answer. In addition, it has been found that these models reached better recalls for what and who questions than where and which questions. This could be explained by the larger number of what and who questions being present in the training set.
Table 6. Question accuracy results of different question generation models.

| Dataset | Model         | Acceptance |
|---------|---------------|------------|
| WebQA   | s2s+Att       | 31%        |
|         | NQG++         | 36%        |
|         | NQG++MP       | 39%        |
|         | BERT-SQG      | 40%        |
|         | mT5           | 42%        |
|         | Single-encQG  | 37%        |
|         | Multi-encQG++ | 45%        |

Table 7. The same question type ratio results of different question generation models.

| Dataset | Model         | What (什么) | Who (谁) | Which (哪个) | Where (哪里) | Avg. |
|---------|---------------|------------|----------|-------------|-------------|------|
| WebQA   | s2s+Att       | 0.625      | 0.636    | 0.599       | 0.393       | 0.563|
|         | NQG++         | 0.708      | 0.682    | 0.416       | 0.643       | 0.612|
|         | NQG++MP       | 0.724      | 0.658    | 0.611       | 0.464       | 0.614|
|         | BERT-SQG      | 0.700      | 0.722    | 0.659       | 0.551       | 0.657|
|         | mT5           | 0.710      | 0.718    | 0.670       | 0.558       | 0.664|
|         | Single-encQG  | 0.647      | 0.586    | 0.671       | 0.557       | 0.615|
|         | Multi-encQG++ | 0.770      | 0.703    | 0.701       | 0.685       | 0.715|

4.6. Error Analysis of Neural Questions

Although our proposed approach with the inclusion of passage compression and multi-encoder sequence-to-sequence neural model augmented with knowledge graph triples has addressed the issue of question generation from a long passage and improved performance, particularly the acceptance and the same question type ratio, which still has around 55% unacceptable questions. The same human annotators have manually analyzed those erroneous questions generated (N = 800 questions) by Multi-encQG++ model in the WebQA dataset and annotated them with one or more deficiencies, such as vagueness, grammatical issues, correct answer issues and question type issues. Final deficiency labels were assigned to a question based on the majority of three annotators indicating deficiency types. An inter-agreement of Fleiss’s was computed for each deficiency type: $k = 0.46$, ungrammatical; $k = 0.45$, vagueness; $k = 0.48$, incorrect answer; $k = 0.42$, wrong question word; and $k = 0.52$, another type, indicating a moderate level of agreement.

Table 8 shows that the most frequent deficiencies of the generated questions were vagueness (38.75%), wrong question word (11.75%), incorrect answer (2.75%) and ungrammatical categories (1.25%). Four main reasons, including multi-word named entity extracting problem, repetition problem, copying wrong context words and less training examples, may cause the poor performance of the model in generating unacceptable questions. We can further elaborate on each of those main causes of deficiencies in the generated questions:

- **Multi-word named entity extracting problem**: only part of the multi-word named entities were extracted from the source text function as a subject of the question, which caused the vague question (see the vague question example in Table 8).

- **Repetition problem**: there are some frequent words that occur in passages, which cause the model to copy these words multiple times in the question (see the ungrammatical question example in Table 8). This issue was also observed in the English sequence-to-sequence learning task [10].

- **Copying wrong context words**: the decoder copied wrong context words from the passage or triples to form the main constituents of a question even if the question word is correct, which led to the incorrect answer problem (see the incorrect answer example in Table 8).
Table 8. Percentage of questions with deficiency generated by Multi-encQG++ model. The passage is compressed, and the target answer is underlined.

| Deficiency                          | Description                                                                 | %  |
|-------------------------------------|-----------------------------------------------------------------------------|----|
| Does not make sense and is vague    | The question is grammatical but too vague to know exactly what it is asking about. Example Passage 2: (鲁智深, an important figure in the novel “Outlaws of the Marsh”, is known as flower monk.) | 38.75 |
|                                    | Tripples: (鲁智深 别名 花和尚), (鲁智深 职业 提辖), (鲁智深 主要成就 拳打镇关西) |    |
|                                    | Reference: 谁是《水浒传》里的花和尚？ (Who is the flower monk in the “Outlaws of the Marsh”?) |    |
|                                    | System: 水浒传里的花是谁？ (Who is the flower from “Heroes of the Marshes”?) |    |
| Wrong question word                | Example Passage 3: 北美洲是世界上湖泊最多的洲同时也是世界上最大的淡水湖区 (North America is the continent with the most lakes in the world and the largest freshwater lake area in the world.) | 11.75 |
|                                    | Tripples: (北美洲,全称,北亚英美洲 ), (北美洲,气候,温带大陆性气候 ), (北美洲,重要城市,纽约) |    |
|                                    | Reference: 世界上哪洲湖泊最多? (Which continent has the most lakes in the world?) |    |
|                                    | System: 世界上最大的淡水湖是什么？ (What is the largest freshwater lake in the world?) |    |
| Incorrect answer                   | The answer to the question is not as the same as human reference. Example Passage 3: 1925年3月，闻一多先生写下了名篇《七子之歌》，其中第五章是“广州湾” (In March 1925, Mr. Wen Yiduo wrote the famous “Song of the Seven Children”, of which the fifth chapter is “Guangzhou Bay”) | 2.75 |
|                                    | Tripples: (闻一多 出生地 湖北省浠水县 ), (闻一多 毕业院校 清华大学 ), (闻一多 职业 现代诗人) |    |
|                                    | (Wen Yiduo birthplace Laishui County, Hubei Province) (Wen Yiduo graduation university) (Wen Yiduo profession modern poet) |    |
|                                    | Reference: 七子之歌是谁写的? |    |
|                                    | Who wrote the Song of the Seven Children? |    |
|                                    | System Question 现代诗人是谁? |    |
|                                    | Who was the modern poet? |    |
| Ungrammatical                      | The question is not a valid Chinese sentence |    |
|                                    | The answer to the question is not as the same as human reference. Example Passage 3: 李贺有“诗鬼”之称，“是与诗圣”之称杜甫、诗仙之称李白”以及“诗佛”王维相属的唐代著名诗人。 Li He is known as a “Poet Ghost”, one of famous poets in Tang Dynasty including Li Bai is known as “Poetic Fairy”, Du Fu is known as “Poet Immortal”, Li P and Wang Wei, “Poet Buddha”. Tripples: (李贺 信仰 道教), (李贺 主要成就 开创“长吉体”诗歌), (李贺 出生地 河南省福昌) | 1.25 |
|                                    | (Li He Religion Taoism), (Li He main achievement initiating “Chang Ji Style” poetry), (Li He birthplace Fuchang Henan) |    |
|                                    | Reference: 文学史上被称为“诗鬼”的作家是谁? |    |
|                                    | Who is known as “poet ghost” in the history of literature? |    |
|                                    | System Question 谁被称为诗鬼称为? |    |
|                                    | Who is known “poet ghost” known? |    |

Fewer training examples (data sparseness): it was found that the model has a tendency to generate questions with a wrong wh-word, such as “what”. It may be due to a lack of training examples, since we observed that there are more “what” questions than the “where” questions in the training set (see the wrong question word example in Table 8).
In addition, we observed that when more relevant sentences were detected in a passage and encoded, it is more challenging to pay attention to the answer and find the correct question words.

5. Conclusions

This article presented a novel automatic Chinese question generation approach, which includes passage compression and augmenting a multi-encoder sequence-to-sequence model with information from a knowledge graph to address major challenges of generating questions from a long sequence of sentences using a classic attention-based sequence-to-sequence model. Our study results indicated that a passage compression step can reduce the complexity of question generation from a long sequence of passage but at the same time decreases performance by losing relevant information about the answer. Thus, augmenting the model with knowledge from a knowledge base to recover answer related information typically lost during the passage compression step improved performance. Our study showed that the multi-encoder sequence-to-sequence neural model with a double attention mechanism worked best for combining compressed passage information, the answer-related triples from the knowledge graph and answer information.

Although this approach has reduced the wrong question word error by 6%, compared to the state-of-the-art BERT-SQL model, we found that the task of Chinese question generation is still challenging if we want the machine generated questions to be as good as the human-generated ones. A total of 17% of system-generated questions with grammatical and semantic correctness failed to match the correct answer or were different from the human question because the model did not pay much attention to relevant context words, which humans use for generating questions (see example passage 3). Liu et al. [49] pointed out that enabling a neural model to learn what to copy is a critical problem. We found most of these copied words are related to the answer’s unique attributes, such as a person’s book (e.g., Wen Yiduo’s “Song of Seven Son”), a singer’s song (e.g., Jay Zhou’s “Coral Sea”) and an artist’s work (e.g., Xu Beihong’s “Running Horse”). These unique attributes can be identified as a clue to help the neural model learn what to copy based on punctuations and grammatical features (e.g., possessive nouns). Thus, a clue prediction mechanism can be developed and embedded into the model [49]. In addition, to solve the multi-word named entity extraction problem, we may implement another mechanism which focuses on copying noun phrases from the context, rather than other unessential words, such as adjectives, adverbs. In conclusion, one of the most significant contributions of this article is the demonstration of how to simplify a text passage and embed knowledge graph information into a multiple encoder sequence-to-sequence neural model, in order to address the complexity of a long sequence of source sentences and improve the overall question generation performance, particularly the same question type ratio. To the best of our knowledge, the proposed approach is novel in the QG research field by having the unique feature of linking a knowledge graph to a neural question generation model, particularly suitable in Chinese due to the language characteristics, and provides potential in generating deeper multiple-choice questions [48], which is what we plan to explore in the future.

In addition, we argue that this approach has the great potential for other language question generation from passages, such as English, since the passage comprehension step introduced in this article can reduce the complexity of a long input sequence for neural sequence-to-sequence learning. But this may cause the loss of some contextual information from which the question may generate since we have found that the NQG++ model performance drops by 3.41 BLEU1 points in the WebQA dataset after passage compression. Thus, it has been found that incorporating the knowledge triples into our proposed multi-encoder sequence-to-sequence neural model improves by 12.57 BLEU points based on the ablation studies. However, different language characteristics, the difficulty level of human questions and the availability of large knowledge graph information are also important factors, which would influence the result. This approach also can be applied to traditional
Chinese or other languages, such as English. Our future work will investigate this approach to other languages, such as English, based on open English knowledge graph databases (e.g., Google Knowledge Graph).

In conclusion, one of the most significant contributions of this article is the demonstration of how to simplify a text passage and embed knowledge graph information into a multi-encoder sequence-to-sequence neural model, to address the complexity of a long sequence of source sentences and improve the overall question generation performance, particularly the same question type ratio. To the best of our knowledge, the proposed approach is novel in the QG research field by having the unique feature of linking a knowledge graph to a neural question generation model, particularly suitable in Chinese due to the language characteristics, and provides potential in generating deeper multiple-choice questions [50], which is what we plan to explore in the future. In addition, large, labeled question and answer datasets are unavailable in the educational domain. Due to the recent development of deep transfer learning research, researchers have investigated self-training and back-training for unsupervised domain adaptation from source to target domain to address this issue [51]. Furthermore, researchers have attempted to improve the neural question generation model performance by adding an answer relevant sentence BERT encoder [52], span-to-span generation mechanism [53]. In future work, we will investigate how to combine some of these technologies with large knowledge bases in the educational domain to generate good quality questions for supporting reading comprehension.

Author Contributions: Conceptualization, M.L.; methodology, M.L.; validation, J.Z. data curation, J.Z.; writing—original draft preparation, M.L.; writing—review and editing, M.L. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities grant number XDJK2019B023 and National Natural Science Foundation grant number 61977054.

Institutional Review Board Statement: “Not applicable” for studies not involving humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from Baidu and are available at http://research.baidu.com with the permission of Baidu.

Acknowledgments: The work was supported by the National Natural Science Foundation (NSF) under grant No. 61977054, the Fundamental Research Funds for the Central Universities (No. XDJK2019B023). Any opinions, findings and conclusions are those of the authors and do not necessarily reflect the views of the above agencies.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Graesser, A.; Person, N. Question Asking during Tutoring. Am. Educ. Res. J. 1994, 31, 104–137. [CrossRef]
2. Rus, V.; Cai, Z.; Graesser, A. Experiments on Generating Questions about Facts. In Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing, Mexico City, Mexico, 18–24 February 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 444–455.
3. Heilman, M.; Smith, N.A. Good Question! Statistical Ranking for Question Generation. In Proceedings of the Annual Conference of North American Chapter of the Association for Computational Linguistics—Human Language Technologies, Los Angeles, CA, USA, 2–4 June 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 609–617.
4. Rus, V.; Wyse, B.; Piwek, P.; Linsean M, S.; Stoyanchev, S.; Moldovan, C. Overview of the First Question Generation Shared Task Evaluation Challenge. In Proceedings of the Third Workshop on Question Generation, Pittsburgh, PA, USA, 6 July 2010.
5. Tang, D.; Duan, N.; Yan, Z.; Zhang, Z.; Zhou, M. Learning to collaborate for question answering and asking. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018.
6. Li, Y.; Duan, N.; Zhou, B.; Chu, X.; Ouyang, W.; Wang, X.; Zhou, M. Visual Question Generation as Dual Task of Visual Question Answering. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Salt Lake City, UT, USA, 18–23 June 2018; pp. 6116–6124.
7. Liu, M.; Rus, V.; Liu, L. Automatic Chinese Factual Question Generation. IEEE Trans. Learn. Technol. 2017, 10, 194–224. [CrossRef]
8. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 2383–2392.

9. Lecun, Y.; Bengio, Y.; Hinton, G. Deep Learning. Nature 2015, 521, 436–444. [CrossRef] [PubMed]

10. See, A.; Liu, P.J.; Manning, C.D. Get to the Point: Summarization with Pointer-Generator Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1073–1083.

11. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2014; pp. 1–15.

12. Du, X.; Shao, J.; Cardie, C. Learning to Ask: Neural Question Generation for Reading Comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1342–1352.

13. Serban, I.V.; García-Durán, A.; Gulcehre, C.; Ahn, S.; Chandar, S.; Courville, A.; Bengio, Y. 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, Berlin, Germany, 7–12 August 2016; pp. 588–598.

14. Zhou, Q.; Yang, N.; Wei, F.; Tan, C.; Bao, H.; Zhou, M. Neural Question Generation from Text: A Preliminary Study. In Proceedings of the International CCF conference on Natural Language Processing and Chinese Computing, Dalian, China, 8–12 November 2017; pp. 662–671.

15. Du, X.; Cardie, C. Harvesting paragraph-level question-answer pairs from wikipedia. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 1907–1917.

16. Zhao, Y.; Ni, X.; Ding, Y.; Ke, Q. Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-attention Networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3901–3910.

17. Zhao, S.; Ng, H.T. Identification and resolution of Chinese zero pronouns: A machine learning approach. In Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing, Prague, Czech Republic, 28–30 June 2007; pp. 1309–1318.

18. Zheng, H.T.; Han, J.; Chen, J.; Sangaihah, A.K. A novel framework for automatic Chinese question generation based on multi-feature neural network model. Comput. Sci. Inf. Syst. 2018, 15, 487–499. [CrossRef]

19. Chan, Y.-H.; Fan, Y.-C. BERT for Question Generation. In Proceedings of the 12th International Conference on Natural Language Generation, Tokyo, Japan, 29 October–1 November 2019; pp. 173–177.

20. Zhu, J.; Xia, Y.; Wu, L.; He, D.; Qin, T.; Zhou, W.; Li, H.; Liu, T.-Y. Incorporating BERT into Neural Machine Translation. In Proceedings of the 8th International Conference on Learning Representation, Addis Ababa, Ethiopia, 26–30 April 2020; pp. 1–18.

21. Goodfellow, I.J.; Warde-Farley, D.; Mirza, M.; Courville, A.; Bengio, Y. Maxout networks. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1319–1327.

22. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the 34th International Conference on Machine Learning, ICML, Sydney, Australia, 6–11 August 2017; pp. 933–941.

23. Xu, B.; Liang, J.; Xie, C.; Liang, B.; Chen, L.; Xiaohidden, Y. CN-DBpedia2: An Extraction and Verification Framework for Enriching Chinese Encyclopedia Knowledge Base. In Proceedings of the Data Intelligence, South Padre Island, TX, USA, 28–30 June 2019; pp. 244–261.

24. Wang, H.; Zhang, X. Baidu Baike. Available online: https://www.baike.baidu.com (accessed on 28 November 2021).

25. Hudong Baike. Available online: https://www.baike.com (accessed on 28 November 2021).

26. Wang, H.; Zhang, X. A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, USENIX Association, OSDI, Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
35. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 20–26 June 2020; pp. 7871–7880.

36. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.Q.; Li, W.; Liu, P. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. J. Mach. Learn. Res. 2020, 21, 1532–4435.

37. Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; Hon, H.W. Unified language model pre-training for natural language understanding and generation. In Proceedings of the In Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 13042–13054.

38. Bao, H.; Dong, L.; Wei, F.; Wang, W.; Yang, N.; Liu, X.; Hon, H.W. Unilmv2: Pseudo-masked language models for unified language model pre-training. In Proceedings of the International Conference on Machine Learning 2020, Shanghai, China, 12–18 July 2020; pp. 642–652.

39. Li, P.; Li, W.; He, Z.; Wang, X.; Cao, Y.; Zhou, J.; Xu, W. Dataset and Neural Recurrent Sequence Labeling Model for Open-Domain Factoid Question Answering. arXiv 2016, arXiv:1607.06275.

40. Xu, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online. 6–11 June 2021; pp. 483–498.

41. Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; Zettlemoyer, L. Multilingual Denoising Pre-training for Neural Machine Translation. Trans. Assoc. Comput. Linguist. MA 2020, 8, 726–742. [CrossRef]

42. Huang, C.; Yin, J.; Hou, F. Text Similarity Measurement Combining Word Semantic Information with TF-IDF Method. Chin. J. Comput. 2011, 34, 856–863. [CrossRef]

43. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 2014, 15, 1929–1958.

44. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; p. 13.

45. Li, S.; Zhao, Z.; Hu, R.; Li, W.; Liu, T.; Du, X. Analogical reasoning on Chinese morphological and semantic relations. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 138–143.

46. t5_in_bert4keras. Available online: https://github.com/bojone/t5_in_bert4keras (accessed on 28 November 2021).

47. Fleiss, J.L. Measuring nominal scale agreement among many raters. Psychol. Bull. 1971, 76, 76–378. [CrossRef]

48. Sun, C. Chinese: A Linguistic Introduction; Cambridge University Press: Cambridge, UK, 2006.

49. Liu, B.; Lai, K.; Zhao, M.; He, Y.; Xu, Y.; Niu, D.; Wei, H. Learning to generate questions by learning what not to generate. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 1106–1118.

50. Stasaski, K.; Hearst, M.A. Multiple Choice Question Generation Utilizing An Ontology. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, Copenhagen, Denmark, 8 September 2018; pp. 303–312.

51. Kulshreshtha, D.; Belfer, R.; Serban, I.V.; Reddy, S. Back-Training excels Self-Training at Unsupervised Domain Adaptation of Question Generation and Passage Retrieval. In Proceedings of the the 2021 Conference on Empirical Methods in Natural Language Processing, Online, 7–11 November 2021; pp. 7064–7078.

52. Back, S.; Kedia, A.; Chinthakindi, S.C.; Lee, H.; Choo, J. Learning to Generate Questions by Learning to Recover Answer-containing Sentences. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 1516–1529.

53. Xiao, D.; Zhang, H.; Li, Y.; Sun, Y.; Tian, H.; Wu, H.; Wang, H. Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In Proceedings of the the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20), Yokohama, Japan, 11–17 July 2020; pp. 3997–4003.