EQG-RACE: Examination-Type Question Generation

Xin Jia, Wenjie Zhou, Xu Sun, Yunfang Wu*
MOE Key Lab of Computational Linguistics, School of EECS, Peking University
{jemmryx, wjzhou013, xusun, wuyf}@pku.edu.cn

Abstract

Question Generation (QG) is an essential component of the automatic intelligent tutoring systems, which aims to generate high-quality questions for facilitating the reading practice and assessments. However, existing QG technologies encounter several key issues concerning the biased and unnatural language sources of datasets which are mainly obtained from the Web (e.g. SQuAD). In this paper, we propose an innovative Examination-type Question Generation approach (EQG-RACE) to generate exam-like questions based on a dataset extracted from RACE. Two main strategies are employed in EQG-RACE for dealing with discrete answer information and reasoning among long contexts. A Rough Answer and Key Sentence Tagging scheme is utilized to enhance the representations of input. An Answer-guided Graph Convolutional Network (AG-GCN) is designed to capture structure information in revealing the inter-sentences and intra-sentence relations. Experimental results show a state-of-the-art performance of EQG-RACE, which is apparently superior to the baselines. In addition, our work has established a new QG prototype with a reshaped dataset and QG method, which provides an important benchmark for related research in future work. We will make our data and code publicly available for further research.

Introduction

Automatic Question Generation (QG) aims at generating grammatical questions for a given text, which is receiving an increasing research interest in the NLP community. Besides enhancing question answering (QA) systems (Tang et al. 2017; Duan et al. 2017; Xu et al. 2019; Zhang and Bansal 2019) and human-machine dialog generation (Wang et al. 2018a), an important purpose of QG is to generate questions of educational materials for reading practice and assessment. To facilitate the instructing process, QG has been investigated for many years (Kunichika et al. 2001; Mitkov 2003; Heilman and Smith 2010).

Existing methods for QG mainly adopt web-extracted QA datasets, such as SQuAD (Rajpurkar et al. 2016), MS-MARCO (Nguyen et al. 2016), NewsQA (Trischler et al. 2016) and CoQA (Reddy, Chen, and Manning 2018), which are not ideal for generating questions in real context. These datasets are either domain-specific or mono-styled (such as news stories for NewsQA and Wikipedia articles for SQuAD and MS-MARCO). Answers in such datasets are often short texts extracted from the context passages and questions are automatically extracted from the Web or generated by crowdworkers. Another dataset, LearningQ (Chen et al. 2018) is an answer-unaware QG dataset, where questions are subjective learner-generated posts extracted from E-learning videos. Different from these datasets, RACE (Lai et al. 2017) is collected from English exams for Chinese students, which is a high-quality examination dataset on reading comprehension in real context. To evaluate learners' cognitive levels in reading, RACE contains articles of diversified genres (such as stories, ads, news, etc.) and questions of various levels. To generate exam-like questions in RACE, models need more advanced abilities of summarization and reasoning.

Most QG systems usually perform sequence-to-sequence generation (Du, Shao, and Cardie 2017; Zhou et al. 2017; Sun et al. 2018; Zhang and Bansal 2019) with attention mechanism. To generate answer-focused questions, features including answer position, POS and NER are used for encoding the contexts (Zhou et al. 2017; Song et al. 2018). Some works also explore Copy or Pointer mechanism (Hosking and Riedel 2019; See, Liu, and Manning 2017; Zhao et al. 2018b) to overcome the OOV problem.

In this paper, we propose Examination-type Question Generation on RACE (EQG-RACE). We clean the RACE dataset and maintain Specific-style questions to construct an examination-type QG dataset. According to our preliminary experiments, current QG models that perform well on SQuAD show much inferior performance on EQG-RACE, indicating the domain-transfer problem of generating examination-type questions.

As a real-world examination data designed by educational experts, there are two main factors that make EQG-RACE more challenging. First, answers are often complete sentences (or long phrases) rather than short text spans contained in the input sequences, making the previous answer tagging method invalid. Second, the context passages are longer and the questions are created through deep reasoning in multiple sentences, making the sequential encoding method like LSTM dysfunction. To address the first issue, we employ a distant-supervised method to find key answer words and key sentences, and then incorporate them into word representa-

* Corresponding author.

Styles	Examples	Train	Dev	Test
Cloze	The last sentence in the passage shows that ___ .	46075	2575	2640
General	What would be the best title for the passage ?	23290	1277	1344
Specific	Why did Tommy's parents send him to a catholic school ?	**18501**	**1035**	**950**

Table 1: Different types of questions in RACE. In constructing our EQG-RACE dataset, we remove Cloze and General style questions and only maintain Specific questions. The right part shows data distribution in the original RACE dataset.

Dataset	Passage.N	Passage.L	Question.N	Question.L	Answer.L	Vocab.N
SQuAD	20958	134.8	97888	11.31	2.91	230399
EQG-RACE	12743	263.3	20486	10.87	6.36	173171

Table 2: Statistics of the most common used QG dataset SQuAD and our reconstructed EQG-RACE. N and L represent the number and average length respectively.

tions. To tackle the second problem and model the reasoning relations within and across sentences, we design an Answer-guided Graph Convolutional Network(AG-GCN) to capture structure information. We conduct a series of experiments on our reconstructed EQG-RACE dataset to explore the performance of existing QG methods and the results validate the effectiveness of our proposed strategies.

Data Construction

In the original RACE dataset, each sample consists of one passage, one question and four options (one right answer and three distractors). For constructing our EQG-RACE dataset, we process the original RACE dataset to suit the QG task.

First, we discard distractors and only maintain the right answer for each sample. Distractors are designed to confuse learners and would introduce noises when generating a grammatical question that can be exactly answered by the right answer.

Second, we filter some types of questions. The questions in RACE can be roughly divided into three types: Cloze, General and Specific. Cloze style questions are in the format of declarative or interrogative sentences with some missing parts to be filled by right answers, which are inappropriate for QG task. General-style questions(e.g. "What would be the best tile for the passage?") are not specific to a particular article, but are widely applicable to any context passages, which can be generated using rule-based methods. Specific-style questions are semantically related to some specific content of the article, which are the focus of our model. Therefore, during reconstructing EQG-RACE we remove Cloze and Gneraral style questions through string matching and hand-crafted rules, and only maintain the Specific style questions. For a better understanding of our data processing, Table 1 gives examples of three style questions as well as their statistics.

As a result, in our EQG-RACE dataset each sample is in the form of <passage, answer, question>. Our task is to automatically generate questions given passage and answer. Two examples of EQG-RACE are given in Table 6 for a case study.

Furthermore, Table 2 gives a descriptive comparison of the most commonly used QG dataset SQuAD and our EQG-RACE. Overall, the scale of EQG-RACE is much smaller than SQuAD (20,486 vs. 97,888 questions). The average passage length of EQG-RACE is 263.3, almost twice the length of SQuAD passage. More importantly, answers in EQG-RACE are generated by human experts and are not simply extracted from the input text. Besides, the answer length of EQG-RACE is more than twice that of SQuAD. These comparisons of textual properties show that EQG-RACE is a more challenging dataset for QG.

The original RACE contains 87,866, 4,887 and 4,934 samples for training, development and testing, respectively. After filtering, the EQG-RACE contains 18,501, 1,035 and 950 <passage, answer, question> triples [1], as shown in Table 1.

Model Description

On the EQG-RACE data, given a passage p and its corresponding answer a, our task aims to generate a grammatical and answer-focused question q:

$$q = \arg\max_{q} P(q|p, a) \tag{1}$$

To handle this challenging task, we propose a unified model by leveraging keywords information, as illustrated in Figure 1. First of all, we annotate passage keywords according to answer information. The input passage is fed into an answer-guided GCN to obtain answer-focused context embedding. Then, features of word embeddings, Keywords tagging embeddings, GCN embeddings and pre-training embeddings are concatenated as the input of a bidirectional LSTM encoder. A gated self-attention mechanism is then applied to the passage hidden states. Upon the above steps, we fuse passage and answer hidden states to obtain answer-aware context representations. Finally, an attention-based decoder generates question in sequence with the help of maxout-pointer mechanism.

Baseline Model

We take the gated self-attention maxout-pointer model (Zhao et al. 2018b) as the baseline model.

For the encoder, we use two-layer bi-directional LSTMs to encode the input passage and get its hidden representations

[1]Data and code available at: https://github.com/jemmryx/EQG-RACE

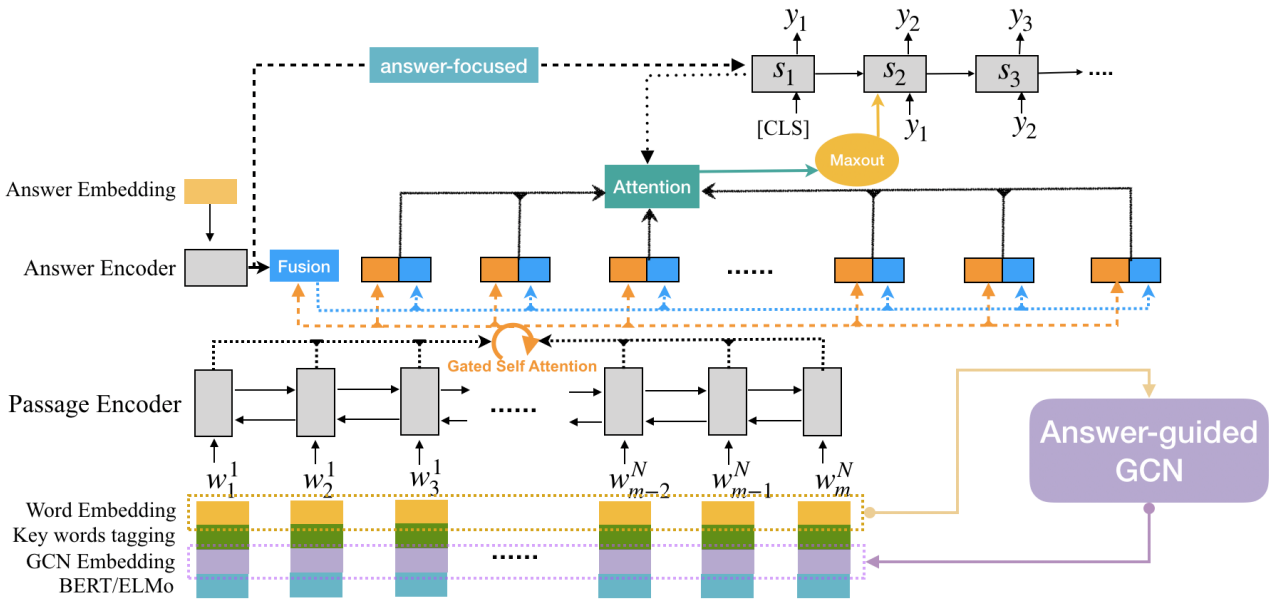

Figure 1: The illustration of our proposed unified model. BERT/ELMo represents using BERT or ELMo pre-trained embedding. (Best viewed in color)

H. At each time step t:

$$h_t^p = LSTM(h_{t-1}^p, e_t^p) \quad (2)$$

where h_t^p and h_{t-1}^p are LSTM hidden states, and e_t^p is word embedding. A gated self-attention mechanism (Wang et al. 2017) is then applied to H to aggregate intra-passage dependencies for $\hat{H}$:

$$s_t^p = H * softmax(H^T W^s h_t^p) \quad (3)$$

$$f_t^p = tanh(W^f[h_t^p, s_t^p]) \quad (4)$$

$$g_t = sigmoid(W^g[h_t^p, s_t^p]) \quad (5)$$

$$\hat{h_t^p} = g_t * f_t^p + (1 - g_t) * h_t^p \quad (6)$$

We obtain self-attention context vector f_t^p by conducting self-matching mechanism on H. Then, a learnable gate g_t is used to balance how much f_t^p and h_t^p will contribute to the encoder output $\hat{H}$.

The decoder is another two-layer uni-directional LSTM. At each time step the decoder state d_{t+1} is generated according to context vector c_t, which aggregates $\hat{H}$ through attention mechanism:

$$\alpha_t = softmax(\hat{H}^\top W_a d_t) \quad (7)$$

$$c_t = \hat{H}\alpha_t \quad (8)$$

$$\hat{d}_t = tanh(W_c[c_t, d_t]) \quad (9)$$

$$d_{t+1} = LSTM([y_t, \hat{d}_t]) \quad (10)$$

The probability of a target word y_t is computed by the maxout-pointer mechanism:

$$p_{vocab} = softmax(W^e d_t) \quad (11)$$

$$p_{copy} = \max_{where\ x_k = y_t} \alpha_{t,k}, \qquad y_t \in X \quad (12)$$

$$p(y_t|y_{<t}) = p_{vocab} * g_p + p_{copy} * (1 - g_p) \quad (13)$$

where X is the vocab of the input sequence and g_p is a trainable parameter to balance p_{vocab} and p_{copy}. p_{vocab} and p_{copy} represent the probability of generating a word from vocab and copying a word from the input sequence respectively.

Keywords Tagging

The questions and answers in RACE are generated by human experts and the answers are not continuous spans in the context, which is different from other common used QG datasets. To this end, the traditional answer tagging methods in QG (Zhou et al. 2017; Yuan et al. 2017; Zhao et al. 2018b) can not be directly used in our task. To introduce answer-focused information into context representations, we employ Keywords tagging methods to locate answer-related words.

Rough Answer Tagging Denote answer words (words occurring in the answer text) as A_t. We first remove stop words in A_t with NLTK to obtain meaningful content words A_f. Then, we match each passage words with A_f and label these matching words with "A" tags.

Key Sentence Tagging Given a context passage, the related questions and answers usually focus on a specific topic which relies on one or several key sentences rather than the full passage. To capture this important information, we find the key sentence in a passage and tag all the words in this sentence with a special label "S". Inspired by the work of Chen and Bansal (2018), for each answer text A_i in EQG-RACE triples, we find the most similar context sentence S_j through:

$$j = \arg\max_t(\text{ROUGE-L}_{recall}(S_t, A_i)) \quad (14)$$

where S_t is the $t-th$ sentence in the input passage.

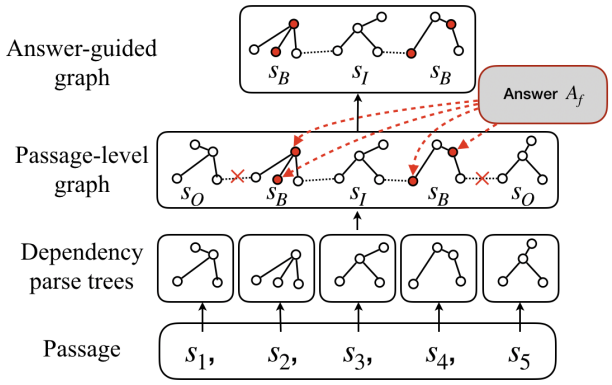

Figure 2: Construction of our Answer-guided graph. After obtaining key answer words A_f, the passage is denoted as: [S_O, S_B, S_I, S_B, S_O]. We delete two S_O sentences since they do not contain answer words and not connect any two S_B.

Note that the answer words tag "A" has a higher priority than "S", which means if a word has both "A" and "S" tags, it will be marked as "A". If a passage word doesn't belong to any answer words nor key sentence words, it will be tagged with "O".

In our experiment, "A", "S" and "O" tags are randomly initialized to 32-dimensional trainable variables and serve as features to enhance context representations. We concatenate Keywords tagging k_t^p with word embeddings e_t^p as feature-enriched input. The Equation 2 can be rewritten as:

$$h_t^p = LSTM(h_{t-1}^p, [e_t^p; k_t^p]) \tag{15}$$

Answer-guided Graph

The questions in RACE are designed to test a learner's understanding ability (such as summarization and attitude analysis). Generating such high-quality questions involves varying cognitive skills and often requires deep reasoning of complex relationships both intra-sentence and inter-sentences. This is different from the existing QG dataset like SQuAD that mainly accommodates factual details in context. As a result, the traditional CNN and RNN methods cannot meet the demands of this difficult task.

To address this issue, we propose an Answer-guided Graph Convolution Network (AG-GCN) to encode passages. As illustrated in Figure 2, the graph construction can be elaborated through the following steps:

- **Step 1**: Implementing dependency parsing for each sentence in the context passage.
- **Step 2**: Linking neighboring sentences' dependency trees by connecting nodes which are at sentence boundaries and next to each other, to build a passage-level dependency parse graph.
- **Step 3**: Retrieving the key answer words A_f for a given answer through Rough Answer Tagging method and removing the "isolated" nodes and their edges from the passage-level graph to construct Answer-guided graph.

After Step 2, we obtain a passage-level dependency parse graph and each passage word corresponds to a node in the graph. However, not every word entitles to generate an answer-focused question, but only the key answer words and its related terms contribute to the QG process. To reduce redundant information and focus on key words, we remove the "isolated" nodes and their edges from the passage graph. Specifically, we divide sentences in the passage into 3 groups according to key answer words A_f: $S_B(Begining)$ represents sentences containing words in A_f; $S_I(Inside)$ represents sentences that connect any two important sentences S_B; others are represented as $S_O(Out)$. We consider the nodes in S_O as "isolated" nodes since they do not contain answer-focused information and can not contribute to the reasoning process between two essential sentences.

The unweighted graph adjacency matrix can be denoted as A, where the weights A_{ij} is 1 if node i is connected with node j otherwise is 0. For the implementation of GCN, we follow the work of Kipf and Welling (2016). We take A and word embeddings E as input, and the encoding process can be formulated as:

$$g_t^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}g_t^l W^l) \tag{16}$$

$$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij} \tag{17}$$

$$\tilde{A} = A + I_N \tag{18}$$

where D is a diagonal matrix and I_N is the identity matrix. The l-th layer of GCN takes the last layer's output g_t^{l-1} as input and g_t^0 is set to E. We adopt a two-layer GCN model.

The output of the last layer g_t^L is fed into a feed-forward layer to obtain the final output g_t^p. We then concatenate it with the word embedding and Keywords tagging embedding as the encoder inputs. Accordingly, Equation 15 can be rewritten as:

$$h_t^p = LSTM(h_{t-1}^p, [e_t^p; k_t^p; g_t^p]) \tag{19}$$

Exploring Pre-training Embeddings

Since the number of EQG-RACE samples is relatively small, the deep neural networks may encounter under-training situations. To solve this problem we take pre-training embeddings p_t^p, such as BERT (Devlin et al. 2019) and ELMo (Peters et al. 2018) embedding, as supplements to the encoder input:

$$h_t^p = LSTM(h_{t-1}^p, [e_t^p; k_t^p; g_t^p; p_t^p]) \tag{20}$$

Here, we obtain the passage encoding h_t^p through a BiLSTM network. Further, we conduct self-attention operation on h_t^p to get high-level representations $\hat{h_t^p}$, by Equation 6.

Passage-answer Fusion

To well capture the inter-dependencies between passage P and answer A, we fuse the answer representation h^A with passage representation $\hat{h_t^p}$ to achieve answer-aware representations as the final encoder outputs:

	EQG-RACE						SQuAD
Previous Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	BLEU-4
Seq2Seq	23.46	10.24	6.39	4.75	23.82	8.57	15.16
Pointer-generator	28.96	14.33	8.82	5.99	30.02	12.26	15.21
HRED	28.69	15.14	9.42	6.16	32.70	12.48	16.43
Transformer	28.93	15.20	9.45	6.25	32.43	13.49	16.50
ELMo-QG	33.95	18.55	11.93	8.23	33.26	14.35	16.75
Our Model							
Unified model + ELMo	**35.10**	**21.08**	**15.19**	**11.96**	**34.24**	**14.94**	–

Table 3: Experimental results of our proposed model comparing with previous methods. We use the source codes available on the web to conduct experiments on our EQG-RACE dataset and SQuAD. Additionally, we apply our **Rough Answer Tagging** strategy to these previous models because the existing answer tagging methods are inappropriate for our EQG-RACE.

$$\tilde{h_t^p} = tanh(W^u[\hat{h_t^p}; h^A; \hat{h_t^p} * h^A; \hat{h_t^p} + h^A]) \quad (21)$$

where h^A is the answer hidden state obtained via a bi-directional LSTM network.

Besides, in the decoding procedure, the first interrogative word is one of the most essential part of the whole generated question. Therefore, instead of using the last hidden state $\hat{h_t^p}$ of the passage encoder, we utilize the answer encoder states h^A as the initialization of the decoder (Kim et al. 2018). Under this setting, the decoder is likely to generate more answer-focused interrogative words.

Experiments and Results

Experimental Settings

In our model, the LSTM hidden sizes of encoder and decoder, the word embedding size and the GCN hidden size are all 300. We set the vocabulary to the most frequent 45,000 words. The maximum lengths of input passage and output question are 400 and 30, respectively. We use pre-trained GloVe embedding as initialization of word embedding and fine-tune it during training. We employ Adam as optimizer with a learning rate 0.001 during training.

For **Unified model + ELMo** settings, we use the pre-trained character-level word embedding from ELMo (Peters et al. 2018) as additional features and also fine-tune it during training. For **Unifed model + BERT**, we replace the ELMo with BERT and keep other settings the same. We use the WordPiece tokenizer to process each word to fit BERT and conduct post-processing to map outputs into normal words.

We utilize Stanford CoreNLP to get dependency parse trees of sentences. The dropout rate of both encoder and GCN is set to 0.3. In decoding, the beam search size is 10.

We evaluate the performance of our models using **BLEU**, **ROUGE-L** and **METEOR**, which are widely used in previous works for QG.

Baseline Models

To compare our model with previous approaches, we re-implement several current neural network-based methods of text generation that have released codes on the web:

- **Seq2seq** (Hosking and Riedel 2019): A RNN sequence-to-sequence model with attention and copy mechanism.
- **Pointer-generator** (See, Liu, and Manning 2017): A LSTM based seq2seq model with pointer mechanism.
- **HRED** (Gao et al. 2018): A seq2seq model with a hierarchical encoder to model the context and to capture both sentence-level and word-level information.
- **Transformer** (Vaswani et al. 2017): A standard transformer-based seq2seq model.
- **ELMo-QG** (Zhang and Bansal 2019): A maxout-pointer model with feature-enriched input which contains answer position, POS, NER and ELMo features.

The most important difference between EQG-RACE and SQuAD is that answers in SQuAD are continuous text spans while answers in EQG-RACE are not. When re-implementing these models, we utilize our proposed Rough Answer Tagging strategy to replace the previous answer tagging method and keep other settings unchanged.

Main Results

The experimental results are shown in Table 3. The traditional text generation models that perform well on SQuAD including Seq2seq, Pointer-generator, and HRED don't work well on EQG-RACE, demonstrating the difficulty and challenging of EQG-RACE. Although the Transformer-based seq2seq model has the ability to capture global information, it does not perform well on this difficult task. Compared with previous approaches, our unified model obtains great improvements over all evaluation metrics. In particular, our method improves a 3.73 BLEU-4 over the best performance ELMo-QG that also utilizes ELMo pre-training embeddings. We achieve a state-of-the-art 11.96 BLEU-4 score on EQG-RACE, which establishes a new benchmark for future research.

Model Analysis

Detailed analysis of the proposed model is presented in Table 4. The maxout-pointer baseline model gets a 4.06 BLEU-4 score without using any answer information. When applying our proposed components, the unified model obtains a much better performance (especially 9.09 BLEU-4 score) on this tough EQG-RACE dataset, which is even better than ELMo-based previous model (8.23 BLEU-4 score). Based on this,

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
Maxout-pointer(no answer information)	27.21	12.51	6.97	4.06	26.57	10.72
Unified model	**36.22**	20.41	13.26	9.09	**35.74**	15.18
- answer-guided-graph	34.15	18.44	11.54	7.43	33.66	14.25
- passage-answer-fusion	34.37	18.89	11.83	7.56	34.62	14.51
- key-sentence-tagging	33.89	18.38	11.54	7.62	34.07	14.28
- rough-answer-tagging	32.74	17.07	10.27	6.32	32.51	13.43
Unified+BERT	34.92	20.73	14.09	10.01	34.63	**15.54**
Unified+ELMo	35.10	**21.08**	**15.19**	**11.96**	34.24	14.94

Table 4: Ablation study of our proposed model on EQG-RACE dataset.

we further apply pre-trained embeddings (BERT and ELMo) and achieve new state-of-the-art results on EQG-RACE.

As shown in the middle part of Table 4, removing each of our proposed modules has varying degrees of decline. Among them, dismantling Rough answer Tagging results in the biggest performance drop (-2.77), because this module highlights the positions directly related to answer which are the most essential information for generating answer-focused questions. The second most influential module is the Answer-guided Graph and removing it will cause the BLEU-4 score to decrease by 1.66. This module provides structure information both within a sentence and between adjacent sentences, which is difficult for LSTM or RNN to capture. Additionally, when removing Fusion or Key Sentence Tagging, the performance shows similar decreases.

Additionally, the pre-trained embeddings improve our model's performance with a large margin. Combining BERT embedding, we obtain a 10.01 BLEU-4 score. We achieve a state-of-the-art result of 11.96 with ELMo embeddings added to our unified model.

Human Evaluation

To assess the quality of generated questions, we conduct human evaluation to compare the baseline maxout-pointer model and our unified model. We randomly select 100 samples and ask three annotators to score them in the scale of [0,2] independently, with the following three metrics:

- **Fluency:** whether a question is grammatical and fluent.
- **Relevancy:** whether the question is semantic relevant to the passage.
- **Answerability:** whether the question can be answered by the right answer.

We then compute the average value of three persons as the final score. As shown in Table 5, our unified model yields higher scores on all three metrics with high Spearman coefficients. Especially, our model obtains an obvious performance gain in **Answerability**, since we incorporate answer information into the neural network in multiple ways.

Case Study

To illustrate our EQG-RACE task and present the output clearly, two real examples of generated questions are shown in Table 6. For the same passage-1, answer-1 and answer-2

Models	Fluency	Relevancy	Answerability
Baseline	1.62	1.45	0.25
Unified	**1.66**	**1.55**	**0.71**
Spearman	0.75	0.57	0.56

Table 5: Human evaluation results of generated questions. The baseline is the maxout-pointer model.

focus on different aspects. The output question-1 is satisfactory because the answer word "*playground*" appears once in the passage, which allows the attention model to focus on its surrounding words to generate a right question. For answer-2, the answer words "*working*", "*talking*" and "*students*" are scattered at several positions which confuse the model's attention.

In passage-2, the keywords of answer-1 ("*easy*", "*get*", "*fish*") appear in several neighbouring sentences. Our Answer-guided graph may capture their inner relationship, enabling our model to produce the right question. As for answer-2, the answer word "*potatoes*" only appears once in the context passage. Our model also generates the right answer but is not as good as the reference question, since the expert leverages external knowledge "*usa is one of the western countries*" that does not appear in the context passage. How to employ knowledge (or common sense) for QG remains an open issue for future research.

Related Work

Generating educational questions for reading practice and assessment is one of the most important applications of QG, which has been proposed and investigated for many years (Rus, Cai, and Graesser 2007; Rus and Lester 2009; Rus et al. 2011; Wang et al. 2018b; Willis et al. 2019). Traditional rule-based methods of QG heavily depend on the quality of handcrafted rules which are time-consuming and laborious. (Wang, Hao, and Liu 2007; Heilman and Smith 2010; Adamson et al. 2013).

Recently, the neural network-based methods for automatically generating questions have achieved great success. Most of these methods treat context passages as inputs and regard questions as targets to perform sequence-to-sequence generation (Du, Shao, and Cardie 2017; Zhao et al. 2018b; Zhou, Zhang, and Wu 2019; Li et al. 2019a; Dong et al. 2019). To incorporate more token features, answer position and other lexical tags such as POS and NER are assimilated into the

Passage-1:	**Passage-2:**
we have twenty minutes' break time after the second class in the morning. look! most of us are playing during the break time. some **students** are on the **playground**. they are playing basketball. oh! a boy is running with the ball. and another is stopping him. they look so cool. and there are some girls watching the game. some **students** are in the classroom. they are **talking**. a few of them are reading and doing homework. look! a girl is looking at the birds in the tree in front of the classroom. she must be thinking of something interesting because she is smiling. what are the teachers doing? some of them are **working** in the office. and some are **talking** with **students**. everyone is doing his or her things, busy but happy!	in some countries, people eat rice every day. sometimes they eat it twice or three times a day for breakfast, lunch and supper. some people do not eat some kinds of meat. muslims, for example, do not eat pork. japanese eat lots of **fish**. they are near the sea. so it is **easy** for them to **get fish**. in the west, such as england and the usa, the most important food is **potatoes**. people there can cook potatoes in many different ways. some people eat only fruit and vegetables. they do not eat meat or **fish** or anything else from animals. they eat food only from plants. they say the food from plants is better for us than meat.
Answer-1: on the **playground**. **Reference-1**: where are the students playing basketball? **Output-1**: where are the students playing basketball?	**Answer-1**: because it is **easy** for them to **get fish**. **Reference-1**: why do japanese eat lots of fish? **Output-1**: why do japanese eat lots of fish?
Answer-2: **working** or **talking** with **students**. **Reference-2**: what are the teachers doing? **Output-2**: what are the students working in the classroom?	**Answer-2**: **potatoes**. **Reference-2**: what is the most important food in some western countries? **Output-2**: what is the most important food in the usa?

Table 6: Case study of our generated questions. We mark the answer words obtained by our Rough Answer Tagging in bold and underline the key sentences obtained by our Key Sentence Tagging strategy.

encoder (Zhou et al. 2017; Sun et al. 2018; Song et al. 2018; Kim et al. 2018; Chen, Wu, and Zaki 2019). To cope with long input contexts, hierarchical architecture has been used to model the contexts and capture sentence-level and word-level importance factors (Gao et al. 2018). Song et al. (2018) propose a matching strategy that firstly select key-sentences from long passages and perform seq2seq generation based on key-sentences. Zhao et al. (2018b) use a self-attention mechanism for the encoder to locate the salient information in a passage. Based on Zhao et al. (2018b)'s work, Zhang and Bansal (2019) apply reinforcement learning to improve the model performance; Nema et al. (2019) utilize an answer encoder to independently encode the answer and fusion passage and answer representations. These methods don't model the structure information which is essential for reasoning among context passages in QG process.

To capture structural information, the Graph Convolutional Networks (GCN) has been employed for both text classification tasks (Peng et al. 2018; Yao, Mao, and Luo 2018; Liu et al. 2018), and generation tasks. Zhao et al. (2018a) utilize GCN to model the dependency between document words to upgrade machine translation. Li et al. (2019b) use a topic interaction graph GCN to generate coherent comments. Chen, Wu, and Zaki (2019) design both static dependency graph and dynamic attention graph GCN to perform question generation. Different from the previous work, we design an Answer-guided GCN to reduce redundant information and focus on answer-related contents.

Conclusion

We present EQG-RACE to automatically generate examination-type questions. We reconstruct the original RACE dataset to suit for question generation and will release this new data. To deal with the discrete answer information in context passages, we propose a Rough Answer and Key Sentence Tagging scheme to locate answer-related contents. Furthermore, we design an Answer-guided graph to capture answer-focused structure information as supplements for seq2seq models. Our integrated model achieves promising results on EQG-RACE and provides a solid benchmark for further research. There is still much room to improvement for this challenging task. We will explore more advanced graph structures to encode the context passages, and employ pre-trained language modeling-based methods on this task.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62076008,61773026) and the Key Project of Natural Science Foundation of China (61936012).

References

Adamson, D.; Bhartiya, D.; Gujral, B.; Kedia, R.; Singh, A.; and Rosé, C. P. 2013. Automatically Generating Discussion Questions. In *AIED*.

Chen, G.; Yang, J.; Hauff, C.; and Houben, G.-J. 2018. LearningQ: A Large-Scale Dataset for Educational Question Generation. In *ICWSM*.

Chen, Y.; Wu, L.; and Zaki, M. J. 2019. Natural Question

Generation with Reinforcement Learning Based Graph-to-Sequence Model. *ArXiv* abs/1910.08832.

Chen, Y.-C.; and Bansal, M. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *ACL*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *NeurIPS*.

Du, X.; Shao, J.; and Cardie, C. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *ACL*.

Duan, N.; Tang, D.; Chen, P.; and Zhou, M. 2017. Question Generation for Question Answering. In *EMNLP*.

Gao, Y.; Bing, L.; Li, P.; King, I.; and Lyu, M. R. 2018. Generating Distractors for Reading Comprehension Questions from Real Examinations. In *AAAI*.

Heilman, M.; and Smith, N. A. 2010. Good Question! Statistical Ranking for Question Generation. In *HLT-NAACL*.

Hosking, T.; and Riedel, S. 2019. Evaluating Rewards for Question Generation Models. In *NAACL-HLT*.

Kim, Y.; Lee, H.; Shin, J.; and Jung, K. 2018. Improving Neural Question Generation using Answer Separation. In *AAAI*.

Kipf, T.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *ArXiv* abs/1609.02907.

Kunichika, H.; Katayama, T.; Hirashima, T.; and Takeuchi, A. 2001. Automated Question Generation Methods for Intelligent English Learning Systems and its Evaluation.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. H. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *EMNLP*.

Li, J.; Gao, Y.; Bing, L.; King, I.; and Lyu, M. R. 2019a. Improving Question Generation With to the Point Context. *ArXiv* abs/1910.06036.

Li, W.; Xu, J.; He, Y.; Yan, S.; Wu, Y.; and Sun, X. 2019b. Coherent Comment Generation for Chinese Articles with a Graph-to-Sequence Model. In *ACL*.

Liu, B.; Zhang, T.; Niu, D.; Lin, J.; Lai, K.; and Xu, Y. 2018. Matching Long Text Documents via Graph Convolutional Networks. *ArXiv* abs/1802.07459.

Mitkov, R. 2003. Computer-aided generation of multiple-choice tests. *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003* 15–.

Nema, P.; Mohankumar, A. K.; Khapra, M. M.; Srinivasan, B. V.; and Ravindran, B. 2019. Let's Ask Again: Refine Network for Automatic Question Generation. *ArXiv* abs/1909.05355.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *ArXiv* abs/1611.09268.

Peng, H.; Li, J.; He, Y.; Liu, Y.; Bao, M.; Wang, L.; Song, Y.; and Yang, Q. 2018. Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN. In Champin, P.; Gandon, F. L.; Lalmas, M.; and Ipeirotis, P. G., eds., *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, 1063–1072. ACM. doi:10.1145/3178876.3186005. URL https://doi.org/10.1145/3178876.3186005.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. S. 2018. Deep Contextualized Word Representations. *ArXiv* abs/1802.05365.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *EMNLP*.

Reddy, S.; Chen, D.; and Manning, C. D. 2018. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics* 7: 249–266.

Rus, V.; Cai, Z.; and Graesser, A. C. 2007. Experiments on Generating Questions About Facts. In *CICLing*.

Rus, V.; and Lester, J. C. 2009. The 2nd Workshop on Question Generation. In *AIED*.

Rus, V.; Wyse, B.; Piwek, P.; Lintean, M. C.; Stoyanchev, S.; and Moldovan, C. 2011. Question Generation Shared Task and Evaluation Challenge - Status Report. In *ENLG*.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL*.

Song, L.; Wang, Z.; Hamza, W.; Zhang, Y.; and Gildea, D. 2018. Leveraging Context Information for Natural Question Generation. In *NAACL-HLT*.

Sun, X.; Liu, J.; Lyu, Y.; He, W.; Ma, Y.; and Wang, S. 2018. Answer-focused and Position-aware Neural Question Generation. In *EMNLP*.

Tang, D.; Duan, N.; Qin, T.; and Zhou, M. 2017. Question Answering and Question Generation as Dual Tasks. *ArXiv* abs/1706.02027.

Trischler, A.; Wang, T.; Yuan, X.; Harris, J.; Sordoni, A.; Bachman, P.; and Suleman, K. 2016. NewsQA: A Machine Comprehension Dataset. In *Rep4NLP@ACL*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*.

Wang, W.; Hao, T.; and Liu, W. 2007. Automatic Question Generation for Learning Evaluation in Medicine. In *ICWL*.

Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated Self-Matching Networks for Reading Comprehension and Question Answering. In *ACL*.

Wang, Y.; Liu, C.; Huang, M.; and Nie, L. 2018a. Learning to Ask Questions in Open-domain Conversational Systems with Typed Decoders. In *ACL*.

Wang, Z.; Lan, A. S.; Nie, W.; Waters, A.; Grimaldi, P. J.; and Baraniuk, R. 2018b. QG-net: a data-driven question generation model for educational content. *Proceedings of the Fifth Annual ACM Conference on Learning at Scale* .

Willis, A.; Davis, G. M.; Ruan, S.; Manoharan, L.; Landay, J. A.; and Brunskill, E. 2019. Key Phrase Extraction for Generating Educational Question-Answer Pairs. *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale* .

Xu, J.; Wang, Y.; Tang, D.; Duan, N.; Yang, P.; Zeng, Q.; Zhou, M.; and Sun, X. 2019. Asking Clarification Questions in Knowledge-Based Question Answering. In *IJCNLP 2019*.

Yao, L.; Mao, C.; and Luo, Y. 2018. Graph Convolutional Networks for Text Classification. In *AAAI*.

Yuan, X.; Wang, T.; Çaglar Gülçehre; Sordoni, A.; Bachman, P.; Zhang, S.; Subramanian, S.; and Trischler, A. 2017. Machine Comprehension by Text-to-Text Neural Question Generation. In *Rep4NLP@ACL*.

Zhang, S.; and Bansal, M. 2019. Addressing Semantic Drift in Question Generation for Semi-Supervised Question Answering. *ArXiv* abs/1909.06356.

Zhao, G.; Li, J. Y.; Wang, L.; Qian, X.; and Fu, Y. 2018a. GraphSeq2Seq: Graph-Sequence-to-Sequence for Neural Machine Translation.

Zhao, Y.; Ni, X.; Ding, Y.; and Ke, Q. 2018b. Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-attention Networks. In *EMNLP*.

Zhou, Q.; Yang, N.; Wei, F.; Tan, C.; Bao, H.; and Zhou, M. 2017. Neural Question Generation from Text: A Preliminary Study. In *NLPCC*.

Zhou, W.; Zhang, M.; and Wu, Y. 2019. Multi-Task Learning with Language Modeling for Question Generation. *ArXiv* abs/1908.11813.