

Enhancing Question Generation with Commonsense Knowledge

Xin Jia[†], Hao Wang[‡], Dawei Yin[‡], Yunfang Wu^{†*}

[†]MOE Key Lab of Computational Linguistics, School of EECS, Peking University
[‡]Baidu Inc., China

{jemmryx, wuyf}@pku.edu.cn
way_wh@yeah.net, yindawei@acm.org

Abstract

Question generation (QG) is to generate natural and grammatical questions that can be answered by a specific answer for a given context. Previous sequence-to-sequence models suffer from a problem that asking high-quality questions requires commonsense knowledge as backgrounds, which in most cases can not be learned directly from training data, resulting in unsatisfactory questions deprived of knowledge. In this paper, we propose a multi-task learning framework to introduce commonsense knowledge into question generation process. We first retrieve relevant commonsense knowledge triples from mature databases and select triples with the conversion information from source context to question. Based on these informative knowledge triples, we design two auxiliary tasks to incorporate commonsense knowledge into the main QG model, where one task is Concept Relation Classification and the other is Tail Concept Generation. Experimental results on SQuAD show that our proposed methods are able to noticeably improve the QG performance on both automatic and human evaluation metrics, demonstrating that incorporating external commonsense knowledge with multi-task learning can help the model generate human-like and high-quality questions.

1 Introduction

Question Generation (QG) has become an essential task for NLP, which aims to generate grammatical and fluent questions for a given context and answer. QG can create question-answer pairs as data augmentation for Question Answering (QA) (Tang et al., 2017; Duan et al., 2017; Zhang and Bansal, 2019). Moreover, it is also useful in education (Heilman and Smith, 2010; Jia et al., 2020a) and business applications (Mostafazadeh et al., 2016), such as creating materials for language beginners, helping build chatbots, etc.

Existing question generation methods can be roughly grouped into two categories. First, rule-based methods utilize handcrafted paradigms to perform declarative-to-interrogative sentence transformations (Heilman and Smith, 2009; Dhole and Manning, 2020), but these methods often consume lots of efforts from domain experts and usually cover limited areas. Second, neural network-based methods typically model the question generation task in a fully data-driven manner (Du et al., 2017; Zhou et al., 2017; Wang et al., 2020a), which has made much progress in recent years.

One key issue, however, is still up in the air: human beings often ask questions with commonsense knowledge that may exist in their brain but not appear in the given context. Take the instance in Table 1 as an example. To generate the human-like high-quality question, one must have the corresponding commonsense knowledge that the “European Parliament” and “Council of the European Union” are of “governing bodies”. Lacking such commonsense knowledge results in unsatisfactory questions that simply copy some words from the source context. This directs us to introduce commonsense backgrounds to bridge the knowledge gap between the given contexts and generated questions.

Actually, previous NLP works have investigated the structured commonsense knowledge to help text generation, such as story generation (Yang et al., 2019; Guan et al., 2019) and response generation (Zhou et al., 2018). They model commonsense knowledge from external databases as additional context through

* Corresponding author.

<p>Context Passage: The European Parliament and the Council of the European Union have powers of amendment and veto during the legislative process.</p>
<p>Reference question: Which governing bodies have legislative veto power?</p>
<p>Generated question by the baseline model: What has the powers of amendment and veto during the legislative process?</p>
<p>Extracted commonsense knowledge: (‘council’, ‘RelatedTo’, ‘governing’) (‘parliament’, ‘Hypernymy’, ‘legislative bodies’)</p>

Table 1: A real example in the training set of SQuAD, which demonstrates the vital effect of commonsense knowledge on QG.

attention mechanism (Zhou et al., 2018; Bai et al., 2019). However, simply employing these methods may not perform well in question generation task. Moreover, there remain two open issues in modeling external knowledge for text generation: 1) existing methods (Zhou et al., 2018; Bai et al., 2019) usually utilize all extracted knowledge triples indiscriminately, which neglects the fact that some triples may not provide useful information, and introduce noises. 2) existing methods simply plug the knowledge triples into encoders, which may not fully leverage the information of these knowledge triples.

We here propose more sophisticated modeling on knowledge triples to help question generation: relevant triples, which cover the knowledge gap between source contexts and generated questions, are selected in the very beginning; furthermore, we not only utilize knowledge triples as additional inputs, but also design auxiliary tasks to help the QG model deeply absorb commonsense knowledge. To the best of our knowledge, we are the first to incorporate structured commonsense knowledge into question generation via a multi-task learning framework.

Specifically, we first retrieve all context-relevant knowledge triples from ConceptNet (Speer et al., 2017) and WordNet (Miller, 1995), and keep the triples where the head concept appears in the context and the tail concept appears in the reference question, i.e., (“council”, “RelatedTo”, “governing”) and (“parliament”, “Hypernymy”, “legislative bodies”) in Table 1.

Then, we design a multi-task learning framework that combines the main QG task and two triple-based auxiliary tasks: Concept Relation Classification and Tail Concept Generation to benefit the question generation process. The two auxiliary tasks can provide useful knowledge information and optimize the parameters of the main QG model.

We conduct extensive experiments on SQuAD dataset, and our proposed model outperforms strong baselines and achieves comparable state-of-the-art performance, demonstrating that incorporating commonsense knowledge with multi-task learning is able to improve the performance of question generation. We will release our data and code for future research.

2 Related Work

Traditionally, QG is tackled by rule-based methods (Heilman and Smith, 2010; Labutov et al., 2015; Dhole and Manning, 2020) that rely heavily on extensive hand-crafted rules. Different from these, neural network-based methods are completely data-driven and trainable in an end-to-end fashion (Du et al., 2017; Zhou et al., 2017; Zhao et al., 2018; Song et al., 2018; Kim et al., 2018; Nema et al., 2019; Zhou et al., 2019a; Zhou et al., 2019b; Jia et al., 2020b; Ko et al., 2020). For better representing the input context, the answer position and token lexical features (e.g. NER, POS and word case) are treated as supplements for the neural encoder (Zhou et al., 2017; Song et al., 2018). Pointer or copy mechanisms (See et al., 2017; Gu et al., 2016; Zhao et al., 2018) are also utilized to overcome the OOV problem in question generation process.

In order to optimize the parameters of QG model, recent works adopt the multi-task learning framework with different auxiliary tasks. Zhou (2019a) use language modeling as a low-level task to provide coherent representations of the input context for the high-level QG task. To improve the accuracy of the start-up word generation, Zhou (2019b) treat question type prediction as an auxiliary task and use the pre-

	SQuAD	knowledge-equipped	pure
ConceptNet	-	12432	-
WordNet	-	41049	-
Whole	75722	46455 (61.3%)	29267 (38.7%)

Table 2: The statistics of retrieved knowledge triples from ConceptNet and WordNet for the training set of SQuAD.

dicted word to initialize the decoding process. Jia (2020b) acquire built-in paraphrase knowledge through back-translation, and introduce paraphrase knowledge into QG process. Different from these works, this paper introduces external commonsense knowledge into QG through multi-task learning framework.

In addition to unstructured knowledge as Jia (2020b) used, many works in text generation utilize structured knowledge from mature database like ConceptNet (Speer et al., 2017). Yang (2019) employ external commonsense knowledge through dynamic memory mechanism to generate more diverse essays. Guan (2019) apply structured commonsense knowledge through multi-source attention to facilitate story comprehension and generate coherent endings. Zhou (2018) incorporate commonsense knowledge graphs through graph attention to create more appropriate and informative responses. Instead of combining knowledge triples into encoding process, our model creatively incorporates commonsense knowledge into QG procedure with two new auxiliary tasks. In this way, the commonsense knowledge can be effectively absorbed by question generation.

3 Knowledge Extraction

To incorporate structured commonsense knowledge into question generation, the most basic step is to extract proper knowledge triples for each training sample. We will describe the details of the knowledge extraction in this section.

In order to obtain more commonsense knowledge, we extract structured knowledge triples from two commonly used databases: ConceptNet and WordNet. ConceptNet is a semantic network composed of triples (h, r, t) denoting that the head concept h has a relation r with tail concept t . WordNet is a lexical database organized in accordance with psycholinguistic theories, where lexicalized concepts are organized by semantic relations (synonymy, hyponymy, etc.). For each sample in the training set of SQuAD, we use each non-stop word in the context passage as a query to retrieve corresponding triples from both ConceptNet and WordNet.

In the process of question generation, only those triples that provide essential knowledge for source contexts and target questions are useful rather than all retrieved triples. Therefore, we design a rule to filter triples: keep only those triples where the head concept h appears in the context passage and the tail concept t appears in the question (in the case of reversed, we swap the head and tail concepts), since these triples can directly provide conversion information between the input context and the output question. For example, in Table 1, we can extract triples like (“council”, “RelatedTo”, “governing”), (“council”, “RelatedTo”, “city”) and (“council”, “Synonymy”, “assembly”) for the word “council”, while we only maintain (“council”, “RelatedTo”, “governing”) since it directly provides the needed information for generating the right question.

From ConceptNet, there are 12,432 training samples that have extracted knowledge triples, and from WordNet there are 41,049 training samples successfully extract knowledge triples. For each training sample, we merge these triples from ConceptNet and WordNet and then remove repeated ones. Finally we obtain 46,455 knowledge-equipped samples and each sample has 1.7 corresponding triples on average, as clearly shown in Table 2.

Accordingly, we divide the origin SQuAD training set into two parts: **commonsense knowledge-equipped samples**: (context passage, answer, question, knowledge triples) and **pure samples**: (context passage, answer, question). We will explain how to use these two parts of data in the following sections.

As shown in Table 3, among these extracted knowledge triples, there are mainly 6 types of relations, where “Synonymy” and “RelatedTo” contribute to the largest proportion, with 41% and 38% respectively.

type	proportion	type	proportion
Synonymy	41%	RelatedTo	38%
IsA	6%	Hypernymy	6%
Hyponymy	3%	Others	6%

Table 3: Relation types of retrieved commonsense knowledge triples. We use “Others” to uniformly denote other relations whose proportion is less than 1%.

4 Model Description

In this section, we will describe our proposed question generation model, as is illustrated in Figure 1. Based on the extracted knowledge triples which can provide commonsense transition information, we incorporate this knowledge into question generation via multi-task learning by employing two triple-based auxiliary tasks.

4.1 Multi-task Learning Framework

For the commonsense knowledge triple $(\mathbf{h}, \mathbf{r}, \mathbf{t})$, where the head concept \mathbf{h} appears in the context passage and tail concept \mathbf{t} appears in the question, it directly provides the conversion information needed for QG. To help the main QG model have a deeper understanding of this commonsense transition, we design two auxiliary tasks: Relation Classification (RC) and Tail Concept Generation (TG). We describe our main QG model and two auxiliary tasks, as well as the unified model in the following parts.

4.1.1 Main Task: QG Baseline Model

Given a context passage p and a specific answer a , QG targets to generate a grammatical question that can be answered by a based on the content of p . We perform sequence-to-sequence generation and adopt the model proposed by Zhang (2019) as our main QG model.

First, we employ a two-layer bi-directional LSTMs as the encoder, which takes feature-enriched embedding e_i as input and outputs a list of hidden representations H :

$$H_i = [\vec{h}_i; \overleftarrow{h}_i] \quad (1)$$

$$\vec{h}_i = \overrightarrow{LSTM}([e_i; \overrightarrow{h}_{i-1}]) \quad (2)$$

$$\overleftarrow{h}_i = \overleftarrow{LSTM}([e_i; \overleftarrow{h}_{i-1}]) \quad (3)$$

$$e_i = [w_i; a_i; n_i; p_i] \quad (4)$$

where w_i, a_i, n_i, p_i respectively represents the embedding of words, answer position (BIO), Name Entity (NER) and Part-of-Speech (POS). For word embedding, we follow the settings of Zhang (2019) and use ELMo (Peters et al., 2018) or BERT (Devlin et al., 2019) to obtain contextualized word representations.

To aggregate long-term dependencies within the context passage, we add a gated self-attention mechanism to the encoder outputs H for \hat{H} :

$$\hat{h}_i^p = g_i * f_i^p + (1 - g_i) * h_i^p \quad (5)$$

We obtain self-attention context vector f_i^p through self-matching mechanism on H and then use a learnable gate g_i to balance how much f_i^p and h_i^p will contribute to the output \hat{H} .

The decoder we used is a two-layer uni-directional LSTM. At each decoding step t , the decoder state s_t is updated dynamically by an attention mechanism on \hat{H} :

$$s_{t+1} = LSTM([y_t, \tilde{s}_t]) \quad (6)$$

$$\tilde{s}_t = \tanh(W^e [c_t; s_t]) \quad (7)$$

$$c_t = \hat{H} \alpha_t, \alpha_t = \text{softmax}(\hat{H}^T W^h s_t) \quad (8)$$

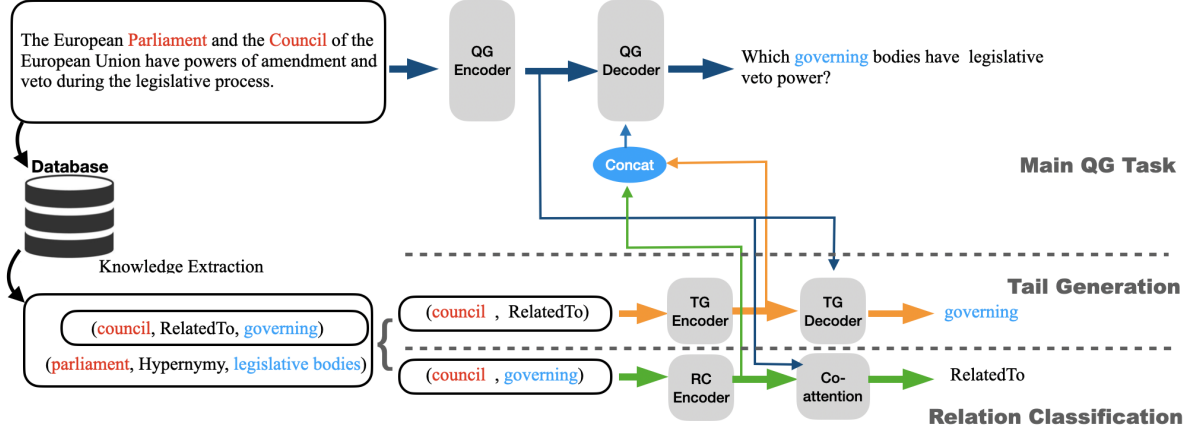


Figure 1: The illustration of our proposed QG framework. First, we conduct knowledge extraction. In the training stage, the two triple-based auxiliary tasks provide commonsense knowledge for the main QG model and the QG model also serves as a context for TG and RC tasks.

For each target word y_t , its probability generated from vocabulary is computed by a maxout neural network and softmax function:

$$\hat{u}_t = \tanh(W^d[c_t; s_t]) \quad (9)$$

$$u_t = [\max\{\hat{u}_{t,2k-1}, \hat{u}_{t,2k}\}]_k \quad (10)$$

$$P_{vocab} = \text{softmax}(W^o u_t) \quad (11)$$

Besides, the pointer mechanism will also be applied to calculate the probability of copying a word from the source context. Finally, the probability distribution is a combination of these two modes with a gate p_g :

$$P(y_t|y_{<t}) = p_g P_{vocab} + (1 - p_g) P_{copy} \quad (12)$$

The training objective is to minimize the negative log likelihood of the target sequence \mathbf{q} :

$$\mathcal{L}_q = -\frac{1}{T_q} \sum_{t=1}^{T_q} \log(P(y_t = \mathbf{q}_t)) \quad (13)$$

4.1.2 Auxiliary Task-1: Relation Classification

This task is designed to predict the correct relationship between the head concept \mathbf{h} and tail concept \mathbf{t} . We use a two-layer bi-directional LSTM to encode the (\mathbf{h}, \mathbf{t}) pair and obtain the hidden representation R . Then we conduct co-attention mechanism between R and the context passage representation \hat{H} to get the co-dependent context \hat{R} :

$$\hat{R} = [\hat{H}; RA^H]A^R \quad (14)$$

$$R = \text{LSTM}([\mathbf{h}; \mathbf{t}]) \quad (15)$$

$$A^H = \text{softmax}((R^T \hat{H})^T) \quad (16)$$

$$A^R = \text{softmax}(R^T \hat{H}) \quad (17)$$

Based on \hat{R} , we use a feed-forward layer f and softmax function to predict the class of relationship:

$$y_r = \text{softmax}(f(\hat{R})) \quad (18)$$

$$\mathcal{L}_r = -\sum_r (\hat{y}_r \log(y_r)) \quad (19)$$

where \mathcal{L}_r is the loss function, and \hat{y}_r is the one-hot label of the relationship class as listed in Table 3.

4.1.3 Auxiliary Task-2: Tail Concept Generation

Correspondingly, given the head concept \mathbf{h} and relationship \mathbf{r} , generating a proper tail concept \mathbf{t} also needs a deep understanding of the commonsense knowledge between them, so we design a second auxiliary task: Tail Concept Generation. Specially, we adopt another two-layer bi-directional LSTM to encode (\mathbf{h}, \mathbf{r}) :

$$T = LSTM([\mathbf{h}; \mathbf{r}]) \quad (20)$$

In the decoding process, we use a uni-directional LSTM to generate tail concept words sequentially based on the head-relation pair. Additionally, QG passage can serve as a background so we add the context representation of QG passage \hat{H} (computed in Equation 5) into tail concept decoding:

$$s_{j+1} = LSTM([y_j, \tilde{s}_j]) \quad (21)$$

$$\tilde{s}_j = \tanh(W^t[c_j; k_j; s_j]) \quad (22)$$

$$c_j = \hat{H}\alpha_j, \alpha_j = \text{softmax}(\hat{H}^T W^h s_j) \quad (23)$$

$$k_j = T\gamma_j, \gamma_j = \text{softmax}(T^T W^k s_j) \quad (24)$$

where y_j refers to the j -th tail concept word, c_j and k_j represent the context of QG passage and head-relation pair, respectively. The word probability calculation is same with QG model. The loss function of tail concept generation is:

$$\mathcal{L}_t = -\frac{1}{T_t} \sum_{j=1}^{T_t} \log(P(y_j = \mathbf{t}_j)) \quad (25)$$

4.1.4 Unified Model

Our unified model combines the main QG model and two auxiliary tasks, as illustrated in Figure 1. In detail, the QG context passage, the head-tail pair and head-relation pair will firstly be encoded by QG encoder, head-tail encoder and head-relation encoder, respectively. Then we concatenate the head-relation and head-tail encoder outputs to obtain the complete knowledge triple representation:

$$K = \text{concat}([T; R]) \quad (26)$$

$$k_t = K\beta_t, \beta_t = \text{softmax}(K^T W^h s_t) \quad (27)$$

We use this representation as additional commonsense context for QG decoding process. Therefore, the Equation 7 can be rewritten as:

$$\tilde{s}_t = \tanh(W^e[c_t; k_t; s_t]) \quad (28)$$

The overall training objective is the combination of the main QG task and two auxiliary tasks:

$$\mathcal{L} = \mathcal{L}_q + \mathcal{L}_r + \mathcal{L}_t \quad (29)$$

As we mentioned above, each knowledge-equipped training sample has 1.7 corresponding triples on average. Since our proposed auxiliary tasks only take one knowledge triple as input at a time, we need to choose one triple among several extracted triples. As shown in Table 3, the types of ‘‘synonymy’’ and ‘‘relatedto’’ have the largest proportions in all extracted triples. We test the effects of prioritizing ‘‘synonymy’’ triples and prioritizing ‘‘relatedto’’ triples. According to experiment results, priority use of ‘‘synonymy’’ is slightly better than the priority use of ‘‘relatedto’’ triples. Based on this, we prefer to use ‘‘synonymy’’ triples in our models.

4.2 Iterative Training Framework

Actually, only 61.3% percent of the training samples have extracted knowledge triples, and our unified model can only be trained on these knowledge-equipped samples. In order to make full use of the remaining pure training samples, we adopt an iterative training framework (ITF) that alternately utilizes knowledge-triple-equipped training data and pure training data.

Our unified model is composed of three parts: QG model, RC model and TG model, where RC and TG models serve as auxiliary tasks to provide commonsense knowledge for QG. During iterative training, the unified model is firstly trained based on knowledge-triple-equipped training data for N steps. Then we switch to the pure training data for another N steps and only update the QG model's parameters and leave the other parameters related to using knowledge triples frozen. In this way, the pure training samples can also contribute to the unified model.

5 Experimental Settings

5.1 Dataset and Metrics

As the most commonly used QG dataset, SQuAD is composed of (passage, answer, question) samples. Follow the setting of Zhang (2019), we split the accessible parts of SQuAD into a training set (75,722 samples), a development set (10,570 samples) and a test set (11,877 samples).

We evaluate the performance of our models using BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004) and METEOR (Denkowski and Lavie, 2014).

5.2 Baseline Models

We compare our method with the following previous works on SQuAD.

- **Rule-based methods:** PCFG-Trans (Heilman and Smith, 2010), Syn-QG (Dhole and Manning, 2020)
- **Pre-trained models:** ACS-QG (Liu et al., 2020), UNILM (Wang et al., 2020b), ERNIE-GEN (Xiao et al., 2020), UNILMv2 (Bao et al., 2020), ProphetNet (Yan et al., 2020)
- **Seq2Seq models:** NQG++ (Zhou et al., 2017), M2S+cp (Song et al., 2018), A-P-Hybrid (Sun et al., 2018), s2s-a-ct-mp-gsa (Zhao et al., 2018), Q-type (Zhou et al., 2019b), sent-Relation (Li et al., 2019), Capture Great Context (Luu et al., 2020), NQG-RL-GS (Chen et al., 2019), QPP&QAP (Zhang and Bansal, 2019), Paraphrase-QG (Jia et al., 2020b)

Besides, in order to present the different performance between our multi-task learning framework of using knowledge triples and previous methods, we also implement the graph attention method (Zhou et al., 2018), which is proposed to facilitate conversation generation task, as a comparing model. This method constructs the retrieved triples into a graph and attentively reads the knowledge triples within each graph through a dynamic graph attention mechanism.

5.3 Implementation Details

Following the settings of Zhang (2019), we tokenize and obtain POS/NER features by Stanford CoreNlp¹. The QG encoder, TG encoder, RC encoder, QG decoder, and TG decoder are all 2-layer LSTMs with the hidden size of 600. We set the probability of dropout to 0.3 for each layer. We use beam search with size of 10 for decoding. In the Iterative Training Framework (ITF), we set N to 3000 and each training mode will be iteratively trained 3 times. In order to reduce the volatility of the training process, we averaged the 5 models closest to the best performing checkpoint on the development set.

The pre-trained ELMo (Peters et al., 2018) word embedding is character-level and we keep it fixed during training. For the BERT (Devlin et al., 2019) version model, we utilize BERT embeddings as the replacement of ELMo. WordPiece tokenizer is applied to tokenize each word and POS/NER tags are also extended to its corresponding word pieces. During inference, we map the word-piece outputs to normal words through post-processing.

¹<http://stanfordnlp.github.io/CoreNLP/>

Categories	Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGH-L	METEOR
Rule-based	PCFG-Trans	28.77	17.81	12.64	9.47	31.68	18.97
	Syn-QG	45.55	30.24	23.84	18.72	-	-
Pre-trained	ACS-QG	52.30	36.70	28.00	22.05	53.25	25.11
	UNILM	-	-	-	23.75	52.04	25.61
	ERNIE-GEN	-	-	-	25.57	53.31	26.89
	UNILMv2	-	-	-	26.30	53.19	27.09
	ProphetNet	-	-	-	26.72	53.79	27.64
Seq2Seq	NQG++	42.36	26.33	18.46	13.51	41.60	18.18
	M2S+cp	-	-	-	13.91	42.72	18.77
	A-P-Hybrid	43.02	28.14	20.51	15.64	-	-
	s2s-a-ct-mp-gsa	44.51	29.07	21.06	15.82	44.24	19.67
	Q-type	43.11	29.13	21.39	16.31	-	-
	Sent-Relation	44.40	29.48	21.54	16.37	44.73	20.68
	Paraphrase-QG	44.32	29.88	22.28	17.21	-	20.96
	Capture Great Context	46.60	31.94	23.44	17.76	45.89	21.56
	NQG-RL-GS	-	-	-	17.94	46.02	21.76
	QPP&QAP	-	-	-	18.65	46.76	22.91
	QG baseline model (with ELMo)	44.99	30.03	22.05	16.70	45.15	21.11
	+ graph attention	45.34	30.18	22.29	17.14	45.04	20.72
	Our Unified model (EMLo)	45.44	30.64	22.63	17.31	46.02	21.58
	QG baseline model (with BERT)	46.12	31.41	23.51	18.19	46.41	21.69
	Our Unified model (BERT)	46.36	31.74	23.91	18.65	46.65	21.84

Table 4: Experimental results of our unified model comparing with previous works

6 Results

6.1 Main Results

The main experimental results are shown in Table 4. For fair comparisons, we divide previous works into three categories: rule-based methods, pre-trained language modeling-based methods and Seq2Seq methods, where rule-based methods and pre-trained models only serve as references and different Seq2Seq models are our comparing methods.

Our unified model (with BERT) outperforms all but one previous Seq2Seq models and obtains comparable results (18.65 BLEU-4 score) with the best QPP&QAP method (Zhang and Bansal, 2019). QPP&QAP method relies on two pre-trained models: question paraphrasing classification and question answering models to provide rewards for policy gradient, and is further equipped with reinforcement learning mechanism, which is more complicated than ours.

Compared with the best performance of rule-based method Syn-QG, our evaluation score of BLEU-4 is very close while our other scores are obviously better.

Although the performance of our model is still far from the methods that are based on pre-training language modeling, our method is much simpler and provides an effective way to use diverse knowledge databases to supplement knowledge triples into QG. Our idea of introducing knowledge into the generation process is completely different from pre-trained models. Actually, pre-training turns out to be a useful strategy for helping models with a large number of parameters based on large-scale unsupervised training corpus (Dong et al., 2019; Liu et al., 2020; Xiao et al., 2020; Bao et al., 2020; Yan et al., 2020). However, it is extremely time-consuming and computationally expensive.

Compared with these, our method is lightweight and only takes several related knowledge triples as additional inputs instead of huge amounts of data.

Besides, directly using top K extracted triples² as additional inputs through graph attention (Zhou et al., 2018) has a slight improvement over the QG baseline model (17.14 vs. 16.70 on BLEU-4). Compared with this traditional method, our proposed multi-task learning framework has better performance, achieving a 0.61 increase over the QG baseline model (with ELMo). Meanwhile, after applying our proposed framework, we also achieve a 0.46 BLEU-4 improvement over the much stronger QG baseline

²We set K to $3*m$ in our experiments, where m represents the number of content words in each paragraph.

model (with BERT), which demonstrates that our proposed methods can indeed improve the performance of question generation.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGH-L	METEOR
QG baseline model	44.47	29.28	21.23	15.90	44.30	20.52
Unified model (w.o. ITF)	44.91	29.96	21.96	16.67	45.48	21.28
-TG	44.30	29.47	21.60	16.32	45.10	20.70
-RC	46.21	30.37	21.95	16.38	44.04	21.10
-TG-RC	45.04	29.71	21.57	16.15	44.84	20.77

Table 5: Ablation studies of our unified model (with ELMo) trained on **knowledge-equipped data**.

6.2 Ablation Study

We perform model variants and conduct ablation tests for better understanding the effect of different components of our model, and the results are shown in Table 5.

Through our Iterative Training Framework (ITF), we can simultaneously utilize the knowledge-equipped training data and pure training data. In order to better displaying the improvement brought by the introduction of external commonsense knowledge alone, we also conduct experiments by training the model only on the **knowledge-equipped training data**. In this case, as shown in Table 5, the QG baseline model (with ELMo) only obtains a 15.90 BLEU-4 score. After applying our proposed components, our unified model boosts the BLEU-4 score to 16.67. That means, with the help of extracted commonsense knowledge and our multi-task learning framework, 61.3% training data (as listed in Table 2) yields very close results to the performance of training on the whole SQuAD dataset (16.70 in Table 4).

To confirm the effect of each component we proposed, we conduct ablation experiments over the unified model based on the **knowledge-equipped training data**. Without the Tail Generation auxiliary task, the performance of our unified model has a drop of 0.35. Besides, the unified model has a performance degradation of 0.29 if removing the Relation Classification task. In the case of removing two auxiliary tasks at the same time (using knowledge triple as an additional input for attention mechanism), the effect of the model will drop by 0.52. These experimental results verify the effectiveness of each auxiliary task and the combination of them.

6.3 Analysis of Auxiliary Tasks

In addition to the main QG task, we also evaluate the performance of two triple-based auxiliary tasks. The Relation Classification task has six categories and its accuracy reaches 66%, which has a 25% increase over the most-frequent-category baseline (41%). For RC task, the relationship between the head and tail concept is generally unique.

On the contrary, Tail Concept Generation is a much difficult task because given head concept and relation, the tail concept is not unique. We use BLEU-1 as evaluate metric and obtain a 6.23 score on this task.

6.4 Human Evaluation

For a text generation task, the automatic metrics like BLEU, ROUGH, and METEOR have limitations to evaluate the quality of generated questions. Therefore, we conduct human evaluation to compare the performance of Unified model (BERT) and QG baseline model (with BERT). We randomly select 100 samples and ask three annotators to score the generated questions of two models, according to: **Relevancy**: whether the question is relevant to the context passage; **Fluency**: whether the question is grammatical and fluent; **Answerability**: whether the question can be answered by the given answer. The rating score is set to [0, 2]. The evaluation results are shown in Table 7. Our unified model receives higher scores on all three metrics. Moreover, the high Spearman correlation coefficients guarantee the validity of our human evaluation results.

Context Passage-1:

Gateway National Recreation Area contains over 26,000 acres (**10,521.83 ha**) in total , most of it surrounded by New York City , including the Jamaica Bay Wildlife Refuge in Brooklyn and Queens , over 9,000 acres (36 km²) of salt marsh , islands , and water , including most of Jamaica Bay . Also

Answer:

10,521.83

Reference:

how large is the gateway national recreation area in **hectares** ?

Baseline:

how many acres contains gateway national recreation area ?

Unified:

how many **hectares** does the gateway national recreation area have?

Context Passage-2:

Australia : The event was held in Canberra , Australian Capital Territory on April 24 , and covered around 16 km of Canberra ’s central areas , from Reconciliation Place to Commonwealth Park . Upon its arrival in Canberra , the Olympic **flame** was presented by Chinese officials to local Aboriginal elder **Agnes Shea** , of the Ngunnawal people . She , in turn

Answer:

Agnes Shea

Reference:

what is the name of the aboriginal elder who received the **torch** from chinese officials ?

Baseline:

who was the olympic flame presented to ?

Unified:

who received the **torch** from chinese officials ?

Table 6: Two real cases in the test set of SQuAD. We bolded the answer and highlight the parts of the passage and the question that require commonsense knowledge conversion. The baseline model refers to QG baseline model (with BERT) and Unified model refers to Unified model (BERT) with ITF.

Model	Relevancy	Fluency	Answerability
baseline/unified	1.39/1.41	1.74/1.80	1.44/1.47
Spearman	0.65	0.80	0.73

Table 7: Human evaluation results.

6.5 Case Study

To clearly display the output questions, two real cases in the test set of SQuAD are shown in Table 6. Generating the right questions needs commonsense knowledge that cannot be directly obtained from the given passage, such as “ha” is the short form of “hectares” in case-1 and “flame” is synonymy with “torch” in case-2. For the baseline model, lacking such commonsense knowledge results in unsatisfactory questions which only copy some words from the passage. Compared with the baseline model, our unified model can better deal with these cases with the help of external commonsense knowledge.

7 Conclusion

In this paper, we propose a new multi-task learning framework to introduce commonsense knowledge into QG. We first extract relevant structured knowledge triples from external databases, ConceptNet and WordNet. Based on these knowledge triples, we design two auxiliary tasks to help the main QG model deeply absorb the commonsense knowledge. Both the automatic and human evaluation results verify the effectiveness of our proposed methods. In the future, we will explore new ways to use multiple knowledge triples simultaneously in the multi-task learning framework. Besides, we may also apply our framework in other text generation tasks, such as conversation generation and story generation.

Acknowledgments

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

References

- Guirong Bai, Shizhu He, Kang Liu, and Jun Zhao. 2019. Variational attention for commonsense knowledge aware conversation generation. In *NLPCC*.
- Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, X. Liu, Yu Wang, Songhao Piao, Jianfeng Gao, M. Zhou, and H. Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *ArXiv*, abs/2002.12804.
- Yu Chen, Lingfei Wu, and Mohammed J. Zaki. 2019. Natural question generation with reinforcement learning based graph-to-sequence model. *ArXiv*, abs/1910.08832.
- Michael J. Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *WMT@ACL*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.
- Kaustubh D. Dhole and Christopher D. Manning. 2020. Syn-qq: Syntactic and shallow semantic rules for question generation. In *ACL*.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NeurIPS*.
- Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *ACL*.
- Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *EMNLP*.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *ArXiv*, abs/1603.06393.
- Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *AAAI*.
- Michael Heilman and Noah A. Smith. 2009. Question generation via overgenerating transformations and ranking.
- Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. In *HLT-NAACL*.
- X. Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2020a. Eqg-race: Examination-type question generation. *ArXiv*, abs/2012.06106.
- Xin Jia, Wenjie Zhou, X. Sun, and Yunfang Wu. 2020b. How to ask good questions? try to leverage paraphrases. In *ACL*.
- Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2018. Improving neural question generation using answer separation. In *AAAI*.
- Wei-Jen Ko, Te-Yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive question generation for high level text comprehension. In *EMNLP*.
- Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *ACL*.
- Jingjing Li, Yifan Gao, Lidong Bing, Irwin King, and Michael R. Lyu. 2019. Improving question generation with to the point context. *ArXiv*, abs/1910.06036.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*.
- B. Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. *Proceedings of The Web Conference 2020*.
- Anh Tuan Luu, Darsh J. Shah, and Regina Barzilay. 2020. Capturing greater context for question generation. In *AAAI*.
- G. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38:39–41.

- Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1802–1813, Berlin, Germany, August. Association for Computational Linguistics.
- Preksha Nema, Akash Kumar Mohankumar, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2019. Let’s ask again: Refine network for automatic question generation. *ArXiv*, abs/1909.05355.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. *ArXiv*, abs/1802.05365.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.
- Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018. Leveraging context information for natural question generation. In *NAACL-HLT*.
- Robyn Speer, J. Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. *ArXiv*, abs/1612.03975.
- Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *EMNLP*.
- Duyu Tang, Nan Duan, Tao Qin, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *ArXiv*, abs/1706.02027.
- Siyuan Wang, Zhongyu Wei, Zhihao Fan, Zengfeng Huang, Weijian Sun, Qi Zhang, and X. Huang. 2020a. Pathqg: Neural question generation from facts. In *EMNLP*.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and M. Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *ArXiv*, abs/2002.10957.
- Dongling Xiao, Han Zhang, Yukun Li, Y. Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. *ArXiv*, abs/2001.11314.
- Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, N. Duan, J. Chen, R. Zhang, and M. Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *ArXiv*, abs/2001.04063.
- Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, and Xu Sun. 2019. Enhancing topic-to-essay generation with external commonsense knowledge. In *ACL*.
- Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. *ArXiv*, abs/1909.06356.
- Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *EMNLP*.
- Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *NLPCC*.
- Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, J. Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*.
- Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019a. Multi-task learning with language modeling for question generation. *ArXiv*, abs/1908.11813.
- Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019b. Question-type driven question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6032–6037, Hong Kong, China, November. Association for Computational Linguistics.