Question Generation Based on Grammar Knowledge and Fine-grained Classification

Yuan Sun[1,2,*] **, Sisi Liu**[1,2] **, Zhengcuo Dan**[1,2] **, Xiaobing Zhao**[1,2,*]
[1]Minzu University of China
[2]National Language Resource Monitoring & Research Center Minority Languages Branch
tracy.yuan.sun@gmail.com, liusisi.s@qq.com, nmzxb_cn@163.com
[*]Corresponding Author: Yuan Sun, Xiaobing Zhao

Abstract

Question generation is the task of automatically generating questions based on given context and answers, and there are problems that the types of questions and answers do not match. In minority languages such as Tibetan, since the grammar rules are complex and the training data is small, the related research on question generation is still in its infancy. To solve the above problems, this paper constructs a question type classifier and a question generator. We perform fine-grained division of question types and integrate grammatical knowledge into question type classifiers to improve the accuracy of question types. Then, the types predicted by the question type classifier are fed into the question generator. Our model improves the accuracy of interrogative words in generated questions, and the BLEU-4 on SQuAD reaches 17.52, the BLEU-4 on HotpotQA reaches 19.31, the BLEU-4 on TibetanQA reaches 25.58.

1 Introduction

Question generation is an essential research direction in natural language generation, which aims to generate a grammatical question based on the given paragraph and answer, and the question can be answered by the given answer. Recently, question generation has received extensive attention in academia and industry. It can automatically generate training data for question answering systems (Du and Cardie, 2018) and reading comprehension tasks (Heilman and Smith, 2010). It can also simulate user questions in the field of education (Du et al., 2017), and guide the machine to ask questions actively in the field of dialogue (Shum et al., 2018), and strengthen human-computer interaction (Mostafazadeh et al., 2016).

In recent years, the research on question generation has achieved great success, but we analyze the questions generated by an advanced answer-aware s2s model (Zhao et al., 2018) and find that some question types and answers did not match.

Paragraph	..., To ensure a space, in 1947, ABC submitted five applications for television station licenses, ...
Answer	1947
Gold	**in what year** did abc submit licenses for 5 television stations ?
Generated	**what** is the name of the space that was used to ensure a space on television station ?

Table 1: English question generation example.

As shown in Table 1, the question type of the gold question is to ask questions about time and use "in what year", but the interrogative word of generated question is "what".

To solve the problem, some studies have proposed to use answer information to improve the interrogative words accuracy of generated questions (Sun et al., 2018; Zhou et al., 2019b; Kang et al., 2019). However, they did not consider the structural and grammatical features of the answers. Some studies classify question types according to the category of interrogative words, and use the first word of the question as the question type. However, we analyze the SQuAD (Rajpurkar et al., 2016) and find that the first word of question is not always an interrogative word, and there is a mismatch between interrogative words and question types. For example, "in what year" is a question about time, but using the classification of interrogative word will classify it into "what" category. Therefore, using interrogative words to classify questions will also introduce wrong information.

To solve above problems, this paper constructs a question type classifier based on a pre-trained language model and a question generator. We integrate the grammatical knowledge of question into the question type classifier, and classify the type according to the inquiry object of the question, we apply this method to English and Tibetan. The grammatical features of Tibetan are different from those of English, for example, interrogative words in English often appear at the beginning of the ques-

Proceedings of the 29th International Conference on Computational Linguistics, pages 6457–6467
October 12–17, 2022.

tions, but in Tibetan, they will appear in multiple positions of questions, and Tibetan interrogative words have ambiguity.

Tibetan is a phonetic writing script, which belongs to the consonant character type. It is divided into two parts: consonants and vowels. The grammatical rules in Tibetan are relatively complex, but there are clear organizational forms and verb changes. Tibetan is composed of syllables, a syllable containing one or up to seven characters (Nuo et al., 2015; Liu et al., 2011), and syllables are separated by an intersyllable marker "·", which is a simple superscripted dot, for example, བཀྲ་ཤིས་ཀྱི་ཚང་ལས་ནི་ཅི་ཞིག་ཡིན། (What is Tashi's major?).

In this paper, we conduct experiments on English and Tibetan datasets respectively, the experimental results show that the BLEU-4 on SQuAD is 17.52, which is 1.21 higher than the experimental result of Zhou et al (Zhou et al., 2019b). BLEU-4 on HotpotQA is 19.31, which is 3.9 higher than the result of Xie et al (Xie et al., 2020). BLEU-2 on TibetanQA is 35.07, which is 9.73 highter than the result of Sun et al (Sun et al., 2021a). The main contributions of this paper are as follows.

(1) Since the classification of questions based on interrogative words will introduce errors, this paper proposes a fine-grained classification method to classify questions according to the object of the question and the lexical features of the answer to guide question generation.

(2) In low-resource languages such as Tibetan, considering the complexity of Tibetan grammar rules and the training data is small, this paper proposes to integrate interrogative grammatical rules to improve the accuracy of question types, and to improve the performance of question generation.

2 Related Work

Question generation mainly adopts two methods, rule-based and neural network-based. The method based on rules and templates is to define some heuristic rules on the syntax tree to convert a sentence into a question (Heilman and Smith, 2010; Mazidi and Nielsen, 2014; Labutov et al., 2015), but the generated questions have limited diversity and poor portability. With the emergence of large-scale and high-quality machine reading comprehension datasets, such as SQuAD (Rajpurkar et al., 2016), MARCO (Nguyen et al., 2016), HotpotQA (Yang et al., 2018) and TriviaQA (Joshi et al., 2017), neural network-based question generation has made great progress.

Existing works are mainly based on seq2seq architecture with attention mechanism and copy mechanism (Du et al., 2017; Zhao et al., 2018; Zhou et al., 2017). To generate answer-related questions, the methods of using the answer position as input or encoding the answer are proposed (Zhou et al., 2017; Song et al., 2018a; Kim et al., 2019). Song et al. (Song et al., 2018b) propose to use a multi-view matching method to calculate the similarity between the answer representation and the hidden layer of the paragraph in three ways, and finally the BLEU-4 of this method reaches 13.98 on SQuAD. Wang et al. (Wang et al., 2020a) introduce a novel hidden answer pivot module to model the hidden answer information to generate questions. Li et al. (Li et al., 2019) jointly model unstructured sentence and structured answer-relevant relation to generate questions. Murakhovs' ka et al. (Murakhovs' ka et al., 2021) train a unified QG model and generate different cognitive levels questions by controlling the answer type. To solve the two problems, the mismatch between the generated question words and answers, and the words copied by the model are far from and irrelevant to the answer. Sun et al. (Sun et al., 2018) propose an answer-focused and position-aware neural question generation model. The final experimental result BLEU-4 reaches 15.64 on SQuAD. To address the challenge of long text on question generation model based on seq2seq, Tuan et al. (Tuan et al., 2020) propose to encode context information in the long text so that model can obtain more information to improve the performance of question generation. Zhao et al. (Zhao et al., 2018) propose a paragraph-level question generation with maxout pointer and gated self-attention, which can effectively utilize paragraph-level context and outperform sentence-level context performance, and finally achieve BLEU-4 of 16.38 on SQuAD. To improve the answerability of the question and the relevance of the original text, Xie et al. (Xie et al., 2020) design three different reward discriminators to calculate reward scores and feed them back to the question generator to improve the fluency, relevance and answerability of generated questions, and finally BLEU-4 reach 15.43 on the HotpotQA (Yang et al., 2018). The above methods improve the relevance between the generated question and the answer, but there is a problem that the generated

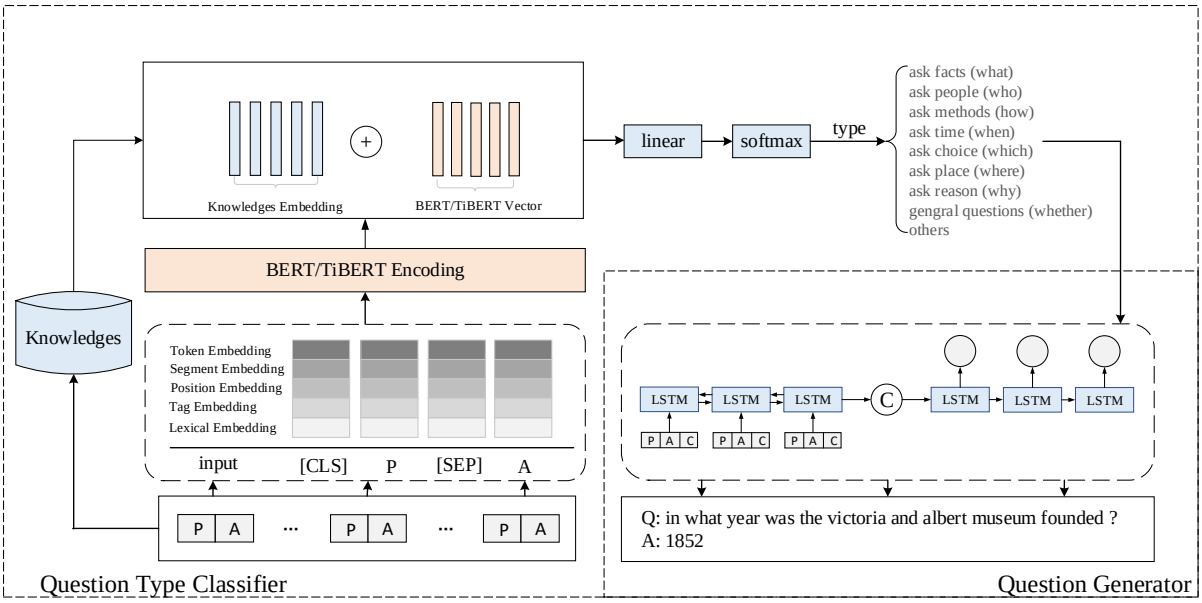

Figure 1: The overall architecture of our model. The question type classifier is used to classify passages and answers to obtain the question types, and then the types are fed into the question generator.

question type and the answer type do not match. To solve the problems, Ma et al. (Ma et al., 2020) propose to integrate sentence-level semantic matching and answer position inference. Zhou et al. (Zhou et al., 2019b) propose a unified model to predict the question type and to generate question. Wang et al. (Wang et al., 2020b) propose use answer-aware initialization module to introduce document and answer to decoder, and design a semantic-rich fusion attention to support decoder. The finally BLEU-4 reach 17.54 on HotpotQA. The above research are applications of question generation on English, there are few related research on Tibetan. Sun et al. (Sun et al., 2021a) construct a Tibetan reading comprehension dataset named TibetanQA, and propose a Tibetan question generation model based on a seq2seq. By introducing an attention mechanism to capture the key semantic information related to the answer and generate an answer-aware question. We follow the method and analyze the generated questions, and find that the accuracy of interrogative words in Tibetan is only 26.03%, which seriously hinders the research on Tibetan question generation. Our method of integrating Tibetan grammar knowledge can significantly improve accuracy of interrogative words in Tibetan question generation.

3 Model Description

The model architecture of this paper is shown in Figure 1. The question type classifier is used to predict the type of target question, and the prediction will be integrated into question generator to guide question generation.

3.1 Fine-grained Question Type Classifier

Different types of questions have different grammar rules, and choosing an accurate interrogative word according to the type of question is the key to generating high-quality questions. To obtain a more accurate question type, this paper divides the datasets according to the objects asked by the question. We use 9 categories to express the question patterns, such as "what" to refer to facts, "who" to refer to people, "how" to refer to methods, "when" to refer to time, "which" to refer to choice, and "where" to refer to a place, "why" to refer to reason, "whether" to refer to general questions and "others" to refer to others.

This paper constructs question type classifier based on BERT (Devlin et al., 2018), and fine-tune BERT with 9 categories data to obtain a classifier. BERT takes a sequence of less than 512 tokens as input and outputs a representation of that sequence. The first token of sequence is always $[CLS]$, which is originally designed for classification tasks, and the other special token $[SEP]$ is used to separate segments. For a given passage P and answer A, the input sequence X as Equation (1).

$$X = ([CLS], P, [SEP], A, [SEP]) \quad (1)$$

In the embedding, we integrate the lexical features of part-of-speech tagging of the answers. The final input embedding $E_f = [E_t; E_s; E_p; E_{tag}; E_l]$,

where E_t is token embedding which contains passage and answer, E_s is segment embedding, E_p is position embedding, E_{tag} is answer tag embedding, and E_l is the embedding of lexical features. The final $V_{[CLS]}$ contains the special categorical embedding, this paper takes $V_{[CLS]}$ as the representation of the sequence, which learns how to represent context and answer information. We add a linear and softmax on top of BERT to predict the probability of question type. The model parameters are shown in Table 2.

Parameters	Values
epoch	10
batch_size	64
pad_size	512
hidden_size	768
learning_rate	5e-5

Table 2: Parameters of question type classifier.

3.2 Question Type Classifier with Grammatical Knowledge

Fine-grained question generation is based on pre-trained language models. Currently, various public multi-language pre-training models such as mBERT (Pires et al., 2019), XLM-RoBERTa (Conneau et al., 2019) and T5 (Raffel et al., 2020) don't contain low-resource languages such as Tibetan. Sun et al. (Sun et al., 2022) construct a Tibetan pre-trained language model named TiBERT. We construct a Tibetan question type classifier based on TiBERT.

Tibetan questions have complex grammatical rules. Different from the situation that English interrogative words often appear at the beginning of sentences, Tibetan interrogative words will appear in various positions of the sentence, and they have ambiguity problems. We will introduce that in detail in section 4. Since Tibetan has the above characteristics, it is difficult to get a correct question type. Thus, this paper integrates Tibetan grammatical knowledges in the question type classifier.

3.3 Question Generator

We follow the Tibetan question generation model based on the Sequence to Sequence Model proposed by Sun et al (Sun et al., 2021a). In the encoder, this paper fuses the output of the question type classifier and the paragraph as the input of the model, and uses a bidirectional LSTM to encode it.

$$h_t = LSTM(h_{t-1}, [P_t, A_t, C_t]) \quad (2)$$

$$H = [\overrightarrow{h_t}, \overleftarrow{h_t}]_{t=1}^{M} \quad (3)$$

where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ represent left-to-right and right-to-left encodings, C_t is the question type predicted by the question type classifier based on paragraph P and answer A, and H is the vector representation encoded by BiLSTM, and then use the self-attention mechanism to process the encoded information. The self-attention mechanism allows the model to dynamically assign weights to different information, and finally obtain an answer-aware contextual representation.

The decoder uses an LSTM with an attention mechanism and a copy mechanism. At time step t, the decoder outputs

$$y_t = LSTM(h_{t-1}, y_{t-1}) \quad (4)$$

where h_{t-1} is the hidden state at time $t-1$, y_{t-1} is the previously output at time $t-1$, and then the model uses the output of the decoder and the contextual representation of the encoder to calculate the attention to obtain the generation probability and copy probability of the word. The probability determines whether the final word is generated from the vocabulary or copied from the context. The model is trained to minimize the negative log-likelihood of the target sequence. The model parameters are shown in Table 3.

4 Influence of Grammar Knowledge on Tibetan Question Generation

4.1 Influence of Interrogative Word Ambiguity

Interrogative words in English will only appear in question in most cases, but there are no fixed interrogative words in Tibetan. The same words may appear as interrogative words or appear as case particles or nouns, therefore, word ambiguity will occur. This paper analyzes interrogative words in Tibetan as follows.

Parameters	Values
param size	41,025,835
epoch	20 in English / 60 in Tibetan
embedding_size	300
hidden_size	300
learning_rate	0.1
dropout	0.3
max_len	400 in English / 1000 in Tibetan

Table 3: Parameters of question generation.

(1) There are no interrogative words indicating “which” and "where" in Tibetan, they usually appear as "གང་" or "གང་ཞིག་", which means “which” or “where” Therefore, the two categories of “which” and “where” are combined into the same category in the Tibetan dataset. However, in addition to appearing as an interrogative word, the "གང་" also appears in the sentence with other meanings. For example, "ཁོ་མོ་རྒྱ་ལ་**གང་**ན་སྐྱན་ལང་**གང་ཞིག**་ལ་འཕྲོ་རྩེས་ཡོད། (Which hospital is she going to go to in the full month?)" The syllable "གང་" appears in this sentence, but it is means “full”, the "གང་ཞིག" in the sentence is the interrogative word, which means “which”.

(2)"སུ་" means "who" in Tibetan, but the syllable "སུ་" also appear in a sentence in the form of a case particle, when the syllable "སུ་" appears as a case particle, the last character of the syllable before the "སུ་" is "ས", which is "སུ་" after removing the root letter, therefore, we use this rule to judge whether "སུ་" appears in the form of interrogative words. For example, "མཚོན་པ་གླུ་རྒྱལ་ནི་སུའི་རྗེས་སུ་བཟུང་ཆྭ་རབ་ཏུ་ཛྷུང་། (Who did the Nagarjuna Bodhisattva become a monk with?)", the interrogative word is "སུའི་", which means “whose”, and indicates “ownership”, the character "ས" appears before the syllable "སུ་", so"སུ་" is a case particle. However, this situation still exists. For example, "འཛམ་བུ་གླིང་ཡིའི་མི་རྣམས་ཀྱི་ཚེ་ལོ་དཔག་མེད་ནས་བྱེ་རྒྱ་བའི་ཐོག་ནས། **སུ་**ཡི་སངས་རྒྱས་བྱོན།

(Which Buddha was born when the life span of people in Nanjiaobu continent was reduced from infinity to 40,000 years?)", here "སུ་" is interrogative word, which means "who", but the syllable "སུ་" appears in the grammatical form of case particle, because the words in front of the "སུ་" is a complete word "སངས་རྒྱས་(Buddha)", and it happens that the last syllable of the words is "ས". In this case, we will give priority to analyzing interrogative words other than "སུ་" , and analyze them in combination with the position of the interrogative words.

4.2 Influence of Tibetan Interrogative Words Position

Interrogative words in English questions usually appear at the beginning of the sentence, it’s syntax is fixed and the training data is large, the model can learn relevant grammar rules and interrogative word position information. But for low-resource languages such as Tibetan, the training data is small and the grammatical rules are complex, so it is difficult for the model to learn relevant knowledge. Therefore, this paper assists the model learning by adding additional grammar knowledge.

This paper analyzes the positions of interrogative words in Tibetan questions as shown in Table 4.

Type	Tibetan question words	Example sentences
Which/Where	གང་དག་/གང་/གང་ཞིག་/གང་ཞིག་/གང་	ཁྱོད་ཀྱི་དཔེ་ཚ་ནི་**གང་དག**་ཡིན། (**Which** are your book?) ཁྱོད་**གང**་ན་ཡོད། (**Where** are you?)
What	ཅི་ཞིག་/ཅི་	བཀྲ་ཤིས་ཀྱི་ཚེད་ལས་ནི་**ཅི་ཞིག**་ཡིན། (**What** is Tashi’s major?)
Who	སུ་	ཁྱོད་**སུ**་ཡིན། (**Who** are you?)
How	ཇི་ཙམ་/ཅི་ཙམ་/གཅད་/ཅི་འདྲ་	ཚོགས་འདུ་འདི་ཀ་སྒྲིག་**ཅི་ཙམ**་བྱེད་དགོས་པ་རེད། (**How** to prepare for this meeting?)
When	ནམ་/ནམ་ཞིག་	པ་ཡུལ་ལ་**ནམ**་ལོག་རྒྱུ་ཡིན། (**When** will you go back to your hometown?)
Why	ཅིའི་ཕྱིར་	དགེ་རྒན་ཀྱིས་**ཅིའི་ཕྱིར**་དེ་ཙམ་བཤད་པ་རེད། (**Why** did the teacher say that?)

Table 4: Position analysis of Tibetan interrogative words.

ཅ་དོག་/སྣ་ཚུ་/དེ་/ཨཚོའུ་/ནི་/ཨ་ཆ་/གང་/དུ་/ཡོད་/།									
(Where is Wucai Lake located?)									
ng	/ ng	/ kg	/ ng	/ up	/ ng	/ rw	/kl	/ve	/xp
E_0	/ E_0	/ E_c	/ E_0	/ E_0	/ E_0	/ E_q	/ E_c	/ E_0	/ E_0

Table 5: An interrogative word position template.

Tibetan interrogatives of "which" and "what" are usually located before modal particles and after verbs. The interrogative words of "who", between the subject and object, not only act as interrogative elements, but also act as predicate elements. The interrogative words of "how" is located between the adverbial and the predicate, with the adverbial in the front and the predicate in the back. The interrogative words of "when" and "why" are usually located before the predicate and after the pronoun or case particle. For the interrogative words of "where", it is located between subject and object, and it also act as predicate. In general, different interrogative words appear in different positions. Therefore, this paper adds lexical features and use rules hard-coded to encode the grammatical knowledge to the Tibetan question type classifier to improve the classification accuracy. A interrogative word position template is as shown in Table 5, where E_q represents the position of interrogative word and E_c represents the position of the case particle.

5 Experiment

5.1 Dataset

We conduct experiments on English dataset SQuAD (Rajpurkar et al., 2016), HotpotQA (Yang et al., 2018) and Tibetan dataset TibetanQA (Sun et al., 2021b). SQuAD is the first large-scale reading comprehension dataset containing natural questions. The articles come from Wikipedia and the dataset is constructed by crowdsourcing to ensure the diversity of questions. HotpotQA contains 113,000 Wikipedia-based question-answer pairs, the answers to the questions are based on multiple supporting documents, and the dataset provides sentence-level supporting fact. TibetanQA contains 1,513 articles and 20,000 question-answer pairs. The articles in the dataset are from yunzang website [1], covering 12 topics such as nature, culture, and education.

This paper counts the distribution of question types in the three datasets, as shown in Table 6. According to the composition of different datasets, this paper divides SQuAD and TibetanQA into 8 categories. The HotpotQA are divided into 9 categories. From Table 6, we can know the distribution of different categories is very uneven, so it is difficult to predict the correct question type.

5.2 Automatic Evaluation

This paper uses BLEU-1, BLEU-2, BLEU-3, BLEU-4 (Papineni et al., 2002) and ROUGE-L (Lin, 2004) to evaluate the performance of the question generation model.

5.3 Experimental Results of Question Generation

We use the question type classifier to classify paragraphs and answers to predict question types, the accuracy of SQuAD is 63.98%, the accuracy of HotpotQA is 55.68% , and the accuracy of TibetanQA is 77.95%. Then, the predicted question types are integrated into question generation, and the final experimental results are shown in Table 7.

To explore the upper bound of our model, and studies the effectiveness of question types on question generation, we use grammatical knowledge to extract the question type and directly fuse it with the paragraph as the input of the question generator.

We compare the performance of our model with previous state-of-the-art models, and the experimental results are shown in Table 7. We briefly introduce these models as follows.

(Zhou et al., 2017): They propose a seq2seq model with attention and copy mechanisms, and use POS and NER tags as lexical features for the encoder.

(Zhao et al., 2018): They propose paragraph-level question generation with maxout pointer mechanism and gated self-attention.

(Zhou et al., 2019a): They incorporate an auxiliary task of language model to the hierarchical multi-task learning structure to help question generation.

(Zhou et al., 2019b): They propose a unified model to predict question type and generate questions.

(Jia et al., 2020): They propose to incorporate paraphrase knowledge into question generation.

(Xie et al., 2020): They propose to optimize question generation with specific rewards.

(Sun et al., 2021a): They propose a seq2seq model with attention and copy mechanisms on Tibetan question generation.

[1] https://www.yongzin.com/

Dataset	What(%)	Who(%)	How(%)	When(%)	Which(%)	Where(%)	Why(%)	Whether(%)	Others(%)
SQuAD	52.45	10.65	10.58	11.73	6.71	4.76	1.37	-	1.76
HotpotQA	34.24	15.15	3.04	10.90	24.98	4.46	0.03	7.10	0.10
TibetanQA	25.22	5.83	21.74	6.48	34.01		0.55	-	5.17

Table 6: The proportion of question types on three datasets.

Datasets	Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
SQuAD	(Zhou et al., 2017)	-	-	-	13.29	-
	(Zhao et al., 2018)	44.51	29.07	21.06	15.82	-
	(Zhou et al., 2019a)	42.80	28.43	21.08	16.23	-
	(Zhou et al., 2019b)	43.11	29.13	21.39	16.31	-
	(Jia et al., 2020)	43.63	29.21	21.79	16.93	-
	Our Model	**47.28**	**31.35**	**23.02**	**17.52**	**46.78**
	Upper Bound	49.02	32.58	24.00	18.31	48.59
HotpotQA	(Zhou et al., 2017)	35.51	22.32	15.94	11.73	32.12
	(Zhao et al., 2018)	38.54	25.09	17.49	13.48	22.45
	(Xie et al., 2020)	37.97	-	-	15.41	35.12
	Our Model	**42.35**	**30.42**	**23.83**	**19.31**	**40.95**
	Upper Bound	45.76	33.04	25.97	21.12	43.50
TibetanQA	(Sun et al., 2021a)	-	25.34	-	-	36.47
	(Sun et al., 2021a) using our metrics	39.18	33.04	28.36	24.77	39.25
	Our Model	**42.45**	**35.07**	**29.64**	**25.58**	**43.28**
	Upper Bound	45.05	37.96	32.30	28.01	44.85

Table 7: Experimental results of question generation on three datasets.

From Table 7, we can know that our model outperforms other models on all metrics, and the effect of the model is significantly improved after the grammar knowledge is integrated into the question generation model. On the English dataset SQuAD, our model achieves 17.52 on BLEU-4, which is 1.21 higher than (Zhou et al., 2019b) and 46.78 on ROUGE-L, and the upper bound of our model can reach 18.31 on BLEU-4. On HotpotQA, our model achieves 19.31 in BLEU-4, which is 3.9 higher than (Xie et al., 2020), and the upper bound of our model can reach 21.12. On TibetanQA, our model achieves 35.07 on BLEU-2, which is 9.73 higher than (Sun et al., 2021a), and 43.28 on ROUGE-L, which is 6.81 higher than (Sun et al., 2021a). The upper bound of our model on TibetanQA can reach 28.1 on BLEU-4.

5.4 Question Type Accuracy of Generated Questions

To further explore the influence of our model on the question type of the generated question, we calculate the accuracy of the interrogative word in the generated questions, as shown in Table 8 and Figure 2, 3, 4.

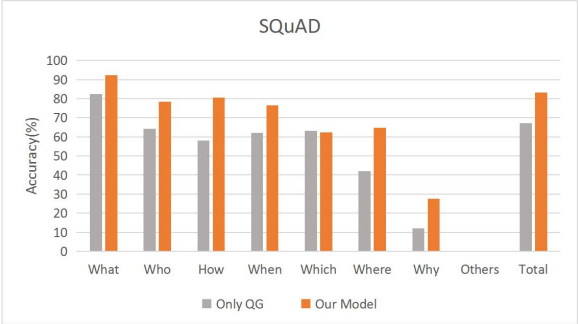

Figure 2: The accuracy on different interrogative words on SQuAD.

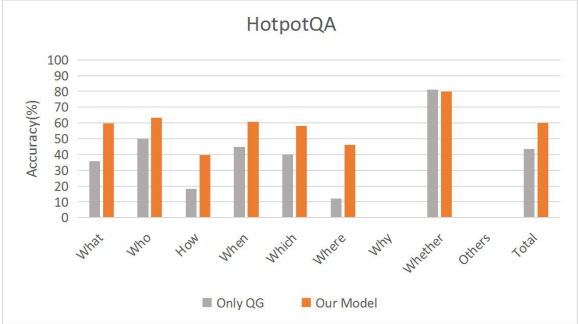

Figure 3: The accuracy on different interrogative words on HotpotQA.

Dataset	Model	What	Who	How	When	Which	Where	Why	whether	Others	Total
SQuAD	Only QG	82.30	64.14	58.03	62.11	63.00	42.08	12.09	-	0	67.15
	Our Model	**92.38**	**78.35**	**80.53**	**76.64**	**62.34**	**64.86**	**27.47**	-	**0**	**83.21**
	Upper Bound	99.55	95.00	99.62	93.77	99.37	98.46	94.51	-	15.00	97.92
HotpotQA	Only QG	35.76	49.90	18.32	44.79	39.95	12.17	0	81.18	0	43.66
	Our Model	**59.70**	**63.15**	**39.69**	**60.85**	**58.2**	**46.03**	**0**	**79.79**	**0**	**59.99**
	Upper Bound	87.80	79.50	71.00	87.04	81.75	83.60	0	98.26	0	89.18
TibetanQA	Only QG	36.11	9.01	0	19.23	1.68		0	-	31.91	26.03
	Our Model	**83.80**	**22.52**	**57.14**	**67.31**	**68.07**		**14.29**	-	**15.51**	**46.88**
	Upper Bound	92.59	17.12	82.43	80.77	92.44		14.29	-	95.51	85.06

Table 8: Accuracy of interrogative words in generated questions on three datasets.

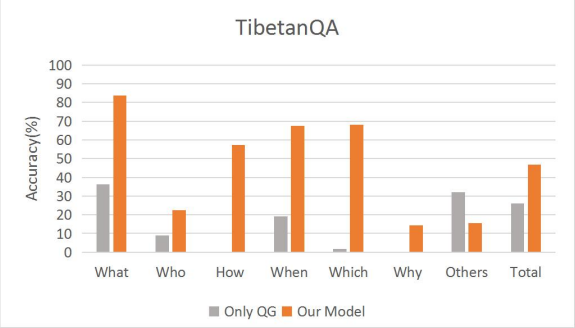

Figure 4: The accuracy on different interrogative words on TibetanQA.

Datasets	Repetition	Incomplete	Accuracy
SQuAD	6.48%	19.18%	87.67%
HotpotQA	5.63%	16.20%	65.50%
TibetanQA	7.38%	39.34%	46.72%

Table 9: Human evaluation results for three datasets.

From Table 8, we can know that the accuracy of interrogative words has been greatly improved. on the SQuAD, the accuracy of interrogative words has increased from 67.15% to 83.21%, and the upper bound of accuracy has reached 97.92%. On the HotpotQA, the accuracy of interrogative words increase from 43.66% to 59.99%, and the upper bound can reach 89.18%. The two categories of "why" and "others" account for 0.03% and 0.1% of the dataset respectively. The small proportion of these two categories makes the model cannot learn the features of these questions. On the TibetanQA, the accuracy of "Only QG" is only 26.03%, the accuracy of our model reaches 46.88%, and the upper bound reaches 85.06%, the accuracy of interrogative words in questions has been greatly improved. We find that the accuracy rate of "who" is low. The main reason is that the grammar of "who" in Tibetan is complex, and the proportion of "who" in TibetanQA is only 5.83%, so it is difficult for the model to learn correct grammar knowledge. In the future work, we will further improve this problem.

Q1:Whether this question is understandable?
A.Yes B.No

Q2:Which of the following errors exists in this question?
A.correct B.Grammatical errors C.repetition
D.incomplete E.ambiguous F.others

Q3:Whether this question is answerable?
A.Yes B.No

Q4:Are the interrogative words and interrogative words in the correct position?
A.Interrogative word and the position of interrogative word are correct
B.Interrogative word is correct but the position of interrogative word is incorrect
C.Interrogative word error
D.Missing Interrogative word

Figure 5: The human evaluation questionnaire.

5.5 Human Evaluation

To further investigate whether the question type classifier improves the quality of generated questions, we conduct human evaluate the final results. We invite six students to analyze the generated questions, three of whom are native speakers of Tibetan and have knowledge of Tibetan linguistics. We randomly select 150 samples from each testing dataset and distribute them equally to these six students. To reduce the subjectivity of human scoring, this paper designs a questionnaire with four questions, as shown in Figure 5. Question 1 evaluates whether the generated question is understandable, if the question is not understandable, we no longer evaluate it for subsequent questions, the question 2 evaluates the fluency of the generated questions, question 3 evaluates whether the generated question is answerable, question 4 evaluates the accuracy of generated interrogative words and the accuracy of the position of interrogative word. The final experimental results of human evaluation are shown in Table 9.

We analyze the obtained questionnaires and discuss the following four findings:

SQuAD	Gold: **how** many permanent objects are located there? Only QG: **what** is the housing of the victoria and albert museum? Our model: **how** many objects does the victoria and albert museum have?
HotpotQA	Gold:**which** tennis player is south african , mariaan de swardt or kateryna bondarenko ? Only QG:**who** was born first , mariaan de swardt or kateryna bondarenko? Our model:**which** tennis player is from south africa , mariaan de swardt or kateryna bondarenko?
TibetanQA	Gold:སྐབས་གཉིས་པའི་འཇམ་གྲྭིང་སྤྱི་རས་དྲ་རྒྱའི་ཚོགས་ཆེན་གྱི་བརྗོད་བྱ་གང་བ་ཞི་**ཅི་ཞེག**ཡིན། (**What** is the theme of the second World Internet Conference?) Only QG:ཚོགས་ཆེན་གྱི་བརྗོད་བྱ་གང་བ་**གཞི་ཚོད་**ཡོད། (**How** many topics are there for the conference?) Our model:ཚོགས་ཆེན་གྱི་བརྗོད་བྱ་གང་བ་གང་བ་བ་གང་བ་**ཅི་ཞེག**ཡིན། (**What** is the theme of the conference?)

Table 10: Questions generated on three datasets.

1. The proportions of unreadable questions generated on TibetanQA, HotpotQA and SQuAD are 18.67%, 5.33% and 2.67%, respectively.

2. From question2, we find that most of the questions generated on the three datasets are fluent, and some generated questions are more suitable than gold questions, this also shows that automatic evaluation is limited. On the other hand, some generated questions have word repetition and incompleteness problems, incomplete questions are mainly due to the lack of keywords, which will also lead to the generated questions being unanswerable.

3. We performe statistics on generated interrogative words and find that, the accuracy of interrogative words on SQuAD and HotpotQA reached 87.67% and 65.50%, the accuracy on TibetanQA reaches 46.72%, which is close to the accuracy of automatic evaluation.

6 Case Analysis

To explore the impact of our model on question generation, Table 10 lists some typical examples. on SQuAD, the interrogative word of gold question is "how", while the interrogative word generated by the baseline model is "what", and the type of question generated by our model is correct. On the HotpotQA, the questions generated by our model are the same as the gold questions, while the interrogative word generated by "Only QG" is "who". On TibetanQA, the interrogative word generated by the baseline model is “how”, and the interrogative word generated by our model is "what", which is the same as the interrogative word of gold question. The above cases show that the question type classifier proposed in this paper can improve the accuracy of question generation.

7 Conclusion

To solve the problem of mismatch between question types and answers in question generation, this paper constructs a question type classifier and a question generator. We classify questions into fine-grained classification and integrate grammar knowledge into question type classifier to improve the accuracy of question types, then, the prediction results of the classifier are fused into the question generator to improve the performance of question generation. To verify the effectiveness of our model, we perform an upper bound analysis on it, we integrate grammar knowledge into the question generator to provide accurate question types to guide the model to generate questions. The final experimental results show that the method proposed in this paper not only improves the accuracy of question words in the generated question, but also improves the quality of the generated question.

Acknowledgment

This work is supported by "National Nature Science Foundation" (61972436), "the Fundamental Research Funds for the Central Universities" (2022QNYL33).

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

X. Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *ArXiv*, abs/1705.00106.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

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

Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2020. How to ask good questions? try to leverage paraphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6130–6140, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Junmo Kang, Haritz Puerto San Roman, and Sung-Hyon Myaeng. 2019. Let me know what to ask: Interrogative-word-aware question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 163–171, Hong Kong, China. Association for Computational Linguistics.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6602–6609.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898, Beijing, China. Association for Computational Linguistics.

Jingjing Li, Yifan Gao, Lidong Bing, Irwin King, and Michael R Lyu. 2019. Improving question generation with to the point context. *arXiv preprint arXiv:1910.06036*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Huidan Liu, Minghua Nuo, Longlong Ma, Jian Wu, and Yeping He. 2011. Tibetan word segmentation as syllable tagging using conditional random field. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, pages 168–177, Singapore. Institute of Digital Enhancement of Cognitive Processing, Waseda University.

Xiyao Ma, Qile Zhu, Yanlin Zhou, and Xiaolin Li. 2020. Improving question generation with sentence-level semantic matching and answer position inferring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8464–8471.

Karen Mazidi and Rodney D. Nielsen. 2014. Linguistic considerations in automatic question generation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 321–326, Baltimore, Maryland. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Lidiya Murakhovs' ka, Chien-Sheng Wu, Tong Niu, Wenhao Liu, and Caiming Xiong. 2021. Mixqg: Neural question generation with mixed answer types. *arXiv preprint arXiv:2110.08175*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Minghua Nuo, Huidan Liu, Congjun Long, and Jian Wu. 2015. Tibetan unknown word identification from news corpora for supporting lexicon-based Tibetan word segmentation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 451–457, Beijing, China. Association for Computational Linguistics.

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.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Heung-Yeung Shum, Xiao-dong He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018a. Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 569–574, New Orleans, Louisiana. Association for Computational Linguistics.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018b. Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 569–574, New Orleans, Louisiana. Association for Computational Linguistics.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939, Brussels, Belgium. Association for Computational Linguistics.

Yuan Sun, Chaofan Chen, Andong Chen, and Xiaobing Zhao. 2021a. Tibetan question generation based on sequence to sequence model. *CMC-COMPUTERS MATERIALS & CONTINUA*, 68(3):3203–3213.

Yuan Sun, Sisi Liu, Chaofan Chen, Zhengcuo Dan, and Xiaobing Zhao. 2021b. Construction of high-quality Tibetan dataset for machine reading comprehension. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 208–218, Huhhot, China. Chinese Information Processing Society of China.

Yuan Sun, Sisi Liu, Junjie Deng, and Xiaobing Zhao. 2022. Tibert: Tibetan pre-trained language model. *arXiv preprint arXiv:2205.07303*.

Luu Anh Tuan, Darsh Shah, and Regina Barzilay. 2020. Capturing greater context for question generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9065–9072.

Bingning Wang, Xiaochuan Wang, Ting Tao, Qi Zhang, and Jingfang Xu. 2020a. Neural question generation with answer pivot. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9138–9145.

Liuyin Wang, Zihan Xu, Zibo Lin, Haitao Zheng, and Ying Shen. 2020b. Answer-driven deep question generation based on reinforcement learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5159–5170, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yuxi Xie, Liangming Pan, Dongzhe Wang, Min-Yen Kan, and Yansong Feng. 2020. Exploring question-specific rewards for generating deep questions. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2534–2546, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.

Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019a. Multi-task learning with language modeling for 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 3394–3399, Hong Kong, China.

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.