Question-type Driven Question Generation

Wenjie Zhou, Minghua Zhang, Yunfang Wu*
Key Laboratory of Computational Linguistics, Ministry of Education
School of Electronics Engineering and Computer Science, Peking University, Beijing, China
{wjzhou013, zhangmh, wuyf}@pku.edu.cn

Abstract

Question generation is a challenging task which aims to ask a question based on an answer and relevant context. The existing works suffer from the mismatching between question type and answer, i.e. generating a question with type *how* while the answer is a personal name. We propose to automatically predict the question type based on the input answer and context. Then, the question type is fused into a seq2seq model to guide the question generation, so as to deal with the mismatching problem. We achieve significant improvement on the accuracy of question type prediction and finally obtain state-of-the-art results for question generation on both SQuAD and MARCO datasets.

1 Introduction

Question generation (QG) can be effectively applied to many fields, including question answering (Duan et al., 2017), dialogue system (Shum et al., 2018) and education. In this paper, we focus on the answer-aware QG, which is to generate a question according to the given sentence and the expected answer.

Recently, the neural-based approaches on QG have achieved remarkable success, by applying large-scale reading comprehension datasets and employing the encoder-decoder framework. Most of the existing works are based on the seq2seq network incorporating attention mechanism and copy mode, which are first applied in Zhou et al. (2017). Later, Song et al. (2018) leverage multi-perspective matching methods, and Sun et al. (2018) propose a position-aware model to put more emphasis on answer-surrounded context words. Both works are trying to enhance the relevance between the question and answer. Zhao et al. (2018) aggregate paragraph-level context to provide sufficient information for question generation. Another direction is to integrate question answering and question generation as dual tasks (Tang et al., 2017). Beyond answer-aware QG, some systems try to generate questions from a text without answer as input (Du and Cardie, 2017; Subramanian et al., 2018).

Despite the progress achieved by the previous work, we found that the types of generated questions are often incorrect. According to experiments on SQuAD, one strong model (Zhou et al., 2017) we replicate only obtains 57.6% accuracy in question type. As we know, question type is vital for question generation, since it determines the question pattern and guides the generating process. If the question type is incorrect, the remained generated sequence would drift far away.

Several works have addressed this issue. Sun et al. (2018) incorporated a question word generation mode to generate question word at each decoding step, which utilized the answer information by employing the encoder hidden states at the answer start position. However, their method did not consider the structure and lexical features of answer. Meanwhile, the way they utilize the question word is not as effective as ours. Hu et al. (2018) proposed a model to generate question based on the given question type and aspect, and their work verifies the effect of question word but fails in the conventional QG task which does not give a question type, since their experimental results show poor performance when trying to generate question types automatically. Our work solves this problem as Section 3.3 shows. Wang et al. (2018) devised a type decoder which combines three type-specific generation distribution (including question type) with weighted sum. However, the results displayed in their paper show that questions in dialogue are far different from questions for reading comprehension, which indicates a gap

*Corresponding author.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 6032–6037,
Hong Kong, China, November 3–7, 2019.

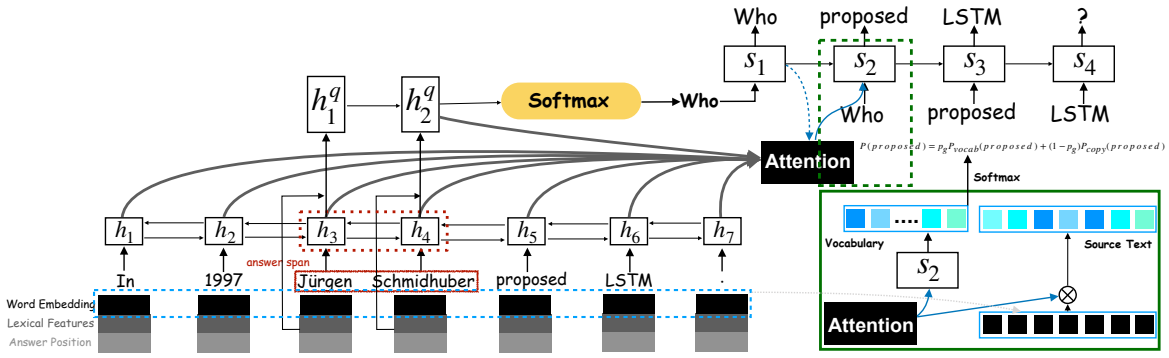

Figure 1: Structure of our unified model

between two tasks.

In this paper, we propose a unified model to predict the question type and to generate questions simultaneously. We conduct experiments on two reading comprehension datasets: SQuAD and MARCO, and obtain promising results. As for the auxiliary task, our unified model boosts the accuracy of question type prediction significantly, by 16.79% on SQuAD and 3.5% on MARCO. For question generation, our model achieves new state-of-the-art results on both datasets, with BLEU-4 16.31 on SQuAD and 21.59 on MARCO.

2 Model Description

The structure of our model is shown in Figure 1. A feature-rich encoder is used to encode the input sentence and corresponding answer that is a span of the sentence. Besides, the answer hidden states are used to predict the type of target question. This prediction will further be used to guide QG with a unified attention-based decoder.

2.1 Feature-rich Encoder

Follow Zhou et al. (2017), we exploit lexical features to enrich the encoder, where features are composed of POS tags, NER tags, and word case. We concatenate the word embedding e_t, answer position embedding a_t and lexical features embedding l_t as input $((x_1, ..., x_T), x_t = [e_t; a_t; l_t])$. Then a bidirectional LSTM is used to produce a sequence of hidden states $(h_1, ..., h_T)$.

2.2 Question Type Prediction

Since different types of questions are various in syntax and semantics, it is essential to predict an accurate type for generating a reasonable question. According to the statistics on SQuAD, 78% of questions in the training set begin with the 7 most common used question words as Table 1 shows. So we divide question types into 8 categories, including 7 question words and an additional type 'others'. It shows that the data distribution on different types is quite unbalanced, suggesting it is a hard task to predict the correct question type.

We use an unidirectional LSTM network to predict the expected question type. Assuming *m+1, ..., m+a* is the index of given answer span in the input sentence, based on the corresponding feature-rich hidden states $(h_{m+1}, ..., h_{m+a})$, we calculate the question type hidden states as follow:

$$h_j^q = LSTM^q([h_{m+j}; l_{m+j}], h_{j-1}^q), j \in [1, a] \quad (1)$$

$$h_0^q = h_T \quad (2)$$

where l_{m+j} is the corresponding lexical features. Besides, to make full use of the feature-rich encoder hidden states, we take the last hidden state as the initial hidden state of $LSTM^q$.

Then, the last output hidden state h_a^q are fed into a softmax layer to obtain the type distribution:

$$P(Q_w) = softmax(W_q h_a^q) \quad (3)$$

$$E^q = -log(P(Q_w^*)) \quad (4)$$

E^q is the loss of question type prediction, Q_w^* is the target type.

2.3 Unified Attention-based Decoder

The conventional attention-based decoder adopts a <BOS> token as the first input word at step 1, while we replace it with the predicted question word Q_w to guide the generation process. At step i, we calculate the decoder hidden state as follow:

$$s_i = LSTM([w_{i-1}; c_{i-1}], s_{i-1}) \quad (5)$$

Further, to fuse the type information in every step of question generation, our model takes h_a^q into consideration while calculating the context

Type	What	Who	How	When	Which	Where	Why	Others
SQuAD	43.26%	9.39%	9.12%	6.26%	4.78%	3.76%	1.37%	21.83%
MARCO	44.29%	1.47%	16.28%	1.52%	1.11%	4.16%	1.88%	29.29%

Table 1: Proportions of each type of questions on two datasets.

vector, i.e. the context vector is conditioned on $(h_1, ..., h_T, h_{T+1})$, where $h_{T+1} = h_a^q$.

$$c_i = \sum_{t=1}^{T+1} \alpha_{it} h_t \tag{6}$$

where α_{it} is calculated via attention mechanism (Bahdanau et al., 2014).

Then, s_i together with c_i will be fed into a two-layer feed-forward network to produce the vocabulary distribution P_{vocab}.

Following (See et al., 2017), the copy mode is used to duplicate words from the source via pointing:

$$P(w_i) = p_g P_{vocab}(w_i) + (1 - p_g) P_{copy}(w_i) \tag{7}$$

where P_{copy} is the distribution of copy mode, $p_g \in [0, 1]$ is a gate to dynamically assign weights.

The loss at step i is the negative log-likelihood of the target word w_i^*. To obtain a combined training objective of two tasks, we add the loss of question type prediction into the loss of question generation to form a total loss function:

$$E^{total} = \frac{1}{K} \sum_{i=1}^{K} -logP(w_i^*) + E^q \tag{8}$$

3 Experiment

3.1 Experiment Settings

Dataset Following the previous works, we conduct experiments on two datasets, SQuAD and MARCO. We use the data released by Zhou et al. (2017) and Sun et al. (2018), there are 86,635, 8,965 and 8,964 sentence-answer-question triples in the training, development and test set for SQuAD, and 74,097, 4,539 and 4,539 sentence-answer-question triples in the training, development and test set for MARCO, respectively. Lexical features are extracted using Stanford CoreNLP.

Implementation Details Our vocabulary is set to contain the most frequent 20,000 words in each training set. Word embeddings are initialized with the pre-trained 300-dimensional Glove vectors, and they are allowed to be fine-tuned during training. The representations of answer position, POS tags, NER tags, and word case are randomly initialized as 32-dimensional vectors, respectively. The feature-rich encoder consists of 2 layers BiLSTM, and the hidden size of all encoders and decoder is set to 512. The cutoff length of the input sequences is set to 10. In testing, we used beam search with a beam size of 12. The development set is used to search the best checkpoint. In order to decrease the volatility of the training procedure, we then average the nearest 5 checkpoints to obtain a single averaged model.

Following the existing work, we use BLUE (Papineni et al., 2002) as the metrics for automatic evaluation, with BLUE-4 as the main metric.

3.2 Upper Bound Analysis

To study the effectiveness of the question type for QG, we make an upper bound analysis. The experimental results are shown in Table 2.

First, we feed the decoder with the original first word of questions, regardless whether the first word is a question word or not. As shown in Table 2, comparing with the baseline model, the performance gets 3.41 and 3.32 points increment on SQuAD and MARCO, respectively, which is a large margin.

Since the beginning words which are not question words compose a large vocabulary while the number of each word is few, it is irrational to train a classifier to predict all types of these beginning words. Therefore, we reduce the question type vocabulary to only question words. In details, if the start of a targeted question is a question word, the corresponding question word will replace the <BOS> to feed into decoder, otherwise we just use the original <BOS> without any replacement. This experiment still gains a lot, as shown in Table 2 with "given the quetion type".

The above experiments verify the magnitude of using the proper question type to guide the generation process, suggesting us a promising direction for QG.

Dataset / Model	SQuAD BLEU-1	SQuAD BLEU-2	SQuAD BLEU-3	SQuAD BLEU-4	MARCO BLEU-1	MARCO BLEU-2	MARCO BLEU-3	MARCO BLEU-4
feature-rich pointer generator (baseline)	41.25	26.76	19.53	14.89	54.04	36.68	26.62	20.15
given the first word of question	47.00	31.82	23.69	18.30	59.51	41.42	30.70	23.47
given the question type	45.56	30.53	22.61	17.38	57.39	39.60	29.29	22.52

Table 2: Upper bound analysis by incorporating question words with different ways.

Dataset / Model	SQuAD BLEU-1	SQuAD BLEU-2	SQuAD BLEU-3	SQuAD BLEU-4	MARCO BLEU-1	MARCO BLEU-2	MARCO BLEU-3	MARCO BLEU-4
NQG++ (Zhou et al., 2017)	-	-	-	13.29	-	-	-	-
question word generate model (Sun et al., 2018)	42.10	27.52	20.14	15.36	46.59	33.46	24.57	18.73
Hybrid model (Sun et al., 2018)	43.02	28.14	20.51	15.64	48.24	35.95	25.79	19.45
Maxout Pointer (sentence) (Zhao et al., 2018)	**44.51**	29.07	21.06	15.82	-	-	-	16.02
Our Model								
Feature-rich pointer generator (Baseline)	41.25	26.76	19.53	14.89	54.04	36.68	26.62	20.15
Question-type driven model (Unified-model)	43.11	**29.13**	**21.39**	**16.31**	**55.67**	**38.16**	**28.12**	**21.59**

Table 3: Experimental results of our model comparing with previous methods on two datasets.

3.3 Results and Analysis

Main Results The experimental results on two datasets are shown in Table 3. By incorporating the question type prediction, our model obtains obvious performance gain over the baseline model, with 1.42 points gain on SQuAD and 1.44 on MARCO. Comparing with previous methods, our model outperforms the existing best methods, achieving new state-of-the-art results on both datasets, with 16.31 on SQuAD and 21.59 on MARCO.

Question Type Accuracy We evaluate different models in terms of Beginning Question Word Accuracy (BQWA). This metric measures the ratio of the generated questions that share the same beginning word with the references which begin with a question word. Table 4 displays the BQWA of two models on both datasets. It shows that our unified model brings significant performance gain in question type prediction.

Further, in Figure 2 we show the accuracy of different question words on both datasets in detail. Our unified model outperforms the baseline for all question types on SQuAD, and all but two types on MARCO.

Model	BQWA on SQuAD	BQWA on MARCO
Baseline	57.62%	76.98%
Unified model	**74.41%**	**80.48%**

Table 4: Experiments of the beginning question word accuracy.

Model Analysis We conduct experiments on different variants of our model, as shown in Table 5. "w/o answer hidden state" takes $(x_{m+1}, ..., x_{m+a})$ as the input of type decoder instead of answer hidden states; "w/o question word replace <BOS>" simply use <BOS> as the first input word of decoder. Experiments on SQuAD verify the effectiveness of our model setting.

Dataset / Model	SQuAD BLEU-1	SQuAD BLEU-2	SQuAD BLEU-3	SQuAD BLEU-4
Unified-model	43.11	29.13	21.39	16.31
w/o answer hidden states	41.91	27.77	20.37	15.54
w/o question word replace <BOS>	41.05	26.93	19.73	15.10

Table 5: Experiments on different settings of our model.

Case Study To show the effect of question words prediction on question generation, Table 6 lists some typical examples.

In the first example, the baseline fails to recognize *Len-shaped* as an adjective, while the unified-model succeeds by utilizing lexical features which are the input of question type prediction layer.

In the second example, the baseline assigns a location type for the generated question based on *American* and *Israel*, it fails to consider the whole answer span. Our unified model resolves it by encoding the answer hidden states sequence as a whole.

The third example shows another typical error, which fails to consider answer surrounding context. In this example the given answer only contains a number *66*. By taking *article 66* into account, we know the question should not be a nu-

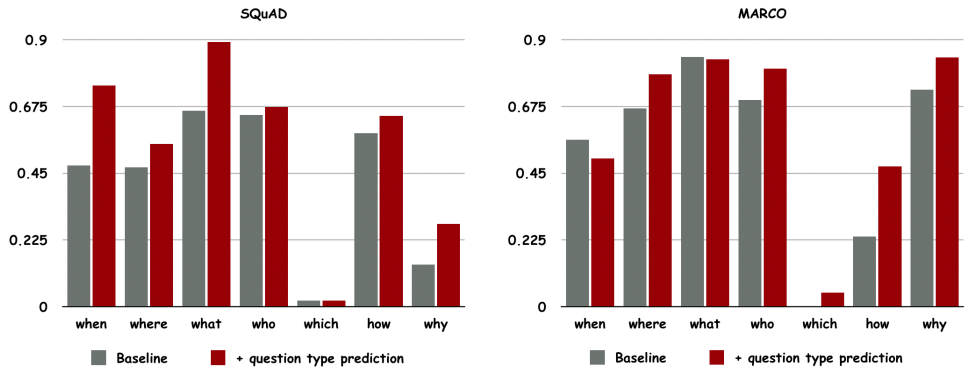

Figure 2: The accuracy on different question words.

merical type. Since the answer hidden states used as the input of type prediction contain surrounding information, our model resolves this problem.

Context: In land plants , chloroplasts are generally lens-shaped, 5–8 m in diameter and 1–3 m thick. **Reference:** How are chloroplasts in land plants usually shaped? **Baseline:** What are chloroplasts generally generally? **Unified-model:** How are chloroplasts in land plants?
Context: In response to American aid to Israel, on October 16, 1973, OPEC raised the posted price of oil by 70%, to $5.11 a barrel. **Reference:** Why did the oil ministers agree to a cut in oil production? **Baseline:** Where did OPEC receive the price of oil by 70%? **Unified-model:** Why did OPEC raised the posted price of oil by 70%?
Context: Article 65 of the agreement banned cartels and article 66 made provisions for concentrations, or mergers, and the abuse of a dominant position by companies. **Reference:** Which article made provisions for concentrations or mergers and the abuse of a dominant position by companies? **Baseline:** How many provisions made provisions for concentrations? **Unified-model:** Which article made provisions for concentrations, or mergers?

Table 6: Examples of generated questions. The underline words are the target answer.

4 Conclusion

In this paper, we discuss the challenge in question type prediction for question generation. We propose a unified model to predict the type and utilize it to guide the generation of question. Experiments on SQuAD and MARCO datasets verify the effectiveness of our model. We improve the accuracy of question type prediction by a large margin, and achieve new state-of-the-art results for question generation.

Acknowledgments

We thank Weikang Li, Xin Jia and Nan Jiang for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (61773026).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2067–2073.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 866–874.

Wenpeng Hu, Bing Liu, Jinwen Ma, Dongyan Zhao, and Rui Yan. 2018. Aspect-based question generation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*.

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, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30*

- *August 4, Volume 1: Long Papers*, pages 1073–1083.

Heung-Yeung Shum, Xiaodong He, and Di Li. 2018. From eliza to xiaoice: challenges and opportunities with social chatbots. *Frontiers of IT & EE*, 19(1):10–26.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. 2018. 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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 569–574.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Adam Trischler, and Yoshua Bengio. 2018. Neural models for key phrase extraction and question generation. In *Proceedings of the Workshop on Machine Reading for Question Answering@ACL 2018, Melbourne, Australia, July 19, 2018*, pages 78–88.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 3930–3939.

Duyu Tang, Nan Duan, Tao Qin, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *CoRR*, abs/1706.02027.

Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2193–2203.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 3901–3910.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *Natural Language Processing and Chinese Computing - 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8-12, 2017, Proceedings*, pages 662–671.