Multi-Task Learning with Language Modeling for 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

This paper explores the task of answer-aware questions generation. Based on the attention-based pointer generator model, we propose to incorporate an auxiliary task of language modeling to help question generation in a hierarchical multi-task learning structure. Our joint-learning model enables the encoder to learn a better representation of the input sequence, which will guide the decoder to generate more coherent and fluent questions. On both SQuAD and MARCO datasets, our multi-task learning model boosts the performance, achieving state-of-the-art results. Moreover, human evaluation further proves the high quality of our generated questions.

1 Introduction

Question generation (QG) receives increasing interests in recent years due to its benefits to several real applications: (1) QG can aid in the development of annotated questions to boost the question answering systems (Duan et al., 2017; Tang et al., 2017); (2) QG enables the dialogue systems to ask questions which make it more proactive (Shum et al., 2018; Colby, 1975); (3) QG can help to generate questions for reading comprehension texts in the education field. In this paper, we focus on answer-aware QG. Giving a sentence and an answer span as input, we want to generate a question whose response is the answer.

Previous work on QG was mainly tackled by two approaches: the rule-based approach and neural-based approach. The neural-based approach receives a booming development due to the release of large-scale reading comprehension datasets like SQuAD (Rajpurkar et al., 2016) and MARCO (Nguyen et al., 2016). Most of the neural approaches on QG employ the encoder-decoder framework, which incorporate attention mechanism to pay more attention to the informative part and copy mode to copy some tokens from the input text (Du et al., 2017; Zhou et al., 2017; Song et al., 2018; Subramanian et al., 2018; Zhao et al., 2018; Sun et al., 2018). To make better use of answer information, Song et al. (2018) leverage multi-perspective matching, and Sun et al. (2018) propose a position-aware model that aims at putting more emphasis on the answer-surrounded context words. Zhao et al. (2018) aggregate paragraph-level information to help QG. Another line of work is to deal with question answering and question generation as dual tasks (Tang et al., 2017; Duan et al., 2017). Some other works try to generate questions from a text without answers as input (Subramanian et al., 2018; Du and Cardie, 2017). Although some progress has been made, there is still much room for improvement for QG.

Multi-task learning is an effective way to improve model expressiveness via related tasks by introducing more data and fruitful semantic information to the model (Caruana, 1998). Many works in NLP have adopted multi-task learning and prove its effectiveness on textual entailment (Hashimoto et al., 2017), keyphrase generation (Ye and Wang, 2018) and document summarization (Guo et al., 2018). To the best of our knowledge, no work attempts to employ multi-task learning for question generation. Although language modeling has been applied to multi-task learning for classification tasks, they are different from our generation task.

In this work, we propose to incorporate language modeling as an auxiliary task to help QG via multi-task learning. We adopt the pointer-generator (See et al., 2017) reinforced with features as the baseline model, which yields state-of-the-art result (Sun et al., 2018). The language modeling task is to predict the next word and the

*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 3394–3399,
Hong Kong, China, November 3–7, 2019.

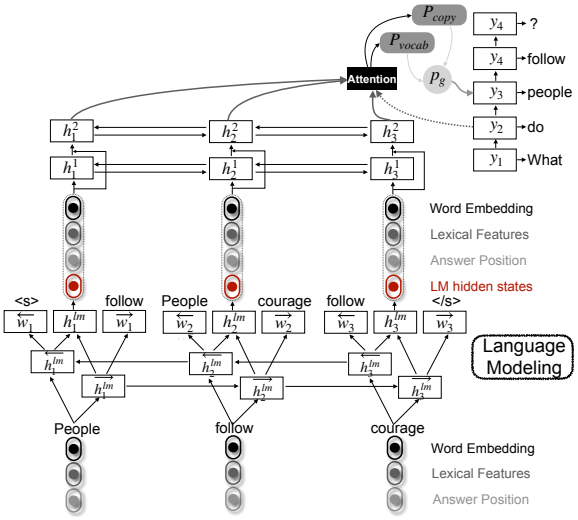

Figure 1: Overall structure of our joint-learning model.

previous word whose input is a plain text without relying on any annotation. The two tasks are then combined with a hierarchical structure, where the low-level language modeling encourages our representation to learn richer language features that will help the high-level network to generate better expressive questions.

We conduct extensive experiments on two reading comprehension datasets: SQuAD and MARCO. We experiment with different settings to prove the efficacy of our multi-task learning model: with/without language modeling and with/without features. Experimental results show that the language modeling consistently yields obvious performance gain over baselines, for all evaluation metrics, including *BLEU*, *perplexity* and *distinct*. Our full model outperforms the existing state-of-the-art results on both datasets, achieving a high BLEU-4 score of 16.23 on SQuAD and 20.88 on MARCO, respectively. We also conduct human evaluation, and our generated questions get higher scores on all three metrics, including $matching$, $fluency$ and $relevance$.

2 Model Description

The baseline model is an attention-based seq2seq pointer-generator reinforced by lexical features, like the work of Sun et al. (2018). In our proposed model, we employ multi-task learning with language modeling as an auxiliary task for QG. The whole structure of our model is shown in Figure 1.

2.1 Feature-enriched Pointer Generator

The feature-rich encoder is a bidirectional LSTM used to produce a sequence of hidden states h_t^L. The encoder takes a sequence of word-and-feature vectors as input ($(x_1, ..., x_T)$), which concatenates the word embedding e_t, answer position embedding a_t and lexical feature embedding l_t ($x_t = [e_t; a_t; l_t]$).The lexical feature is composed of POS tags, NER tags, and word case.

The attention-based decoder is another unidirectional LSTM, which is conditioned on the previous decoder state s_{i-1}, decoded word w_{i-1}, and context vector c_{i-1} which is generated via attention mechanism (Bahdanau et al., 2014):

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

Further, a two-layer feed-forward network is used to produce the vocabulary distribution P_{vocab}.

The pointer generator (See et al., 2017) incorporates a copy mode $P_{copy}(w)$, which allows copying words from the source text via pointing. The final probability distribution is to combine both modes with a generation probability $p_g \in [0, 1]$:

$$P(w) = p_g P_{vocab}(w) + (1 - p_g) P_{copy}(w) \quad (2)$$

The model is trained to minimize the negative log-likelihood of the target sequence. We denote this loss as E.

2.2 Language Modeling

The language model is to predict the next word and previous word in the sequence with a forward LSTM and a backward LSTM, respectively. First, we feed the input sequence into a bidirectional LSTM to get the hidden representations h_t^{lm}.

Then, these states are fed into a softmax layer to predict the next and the previous word:

$$P^{lm}(w_{t+1}|w_{<t+1}) = softmax(W_f \overrightarrow{h_t^{lm}}) \quad (3)$$

$$P^{lm}(w_{t-1}|w_{>t-1}) = softmax(W_b \overleftarrow{h_t^{lm}}) \quad (4)$$

The training objective is to minimize the loss function which is defined as the average of the negative log-likelihood of the next word and the previous word in the sequence:

$$E^{lm} = -\frac{1}{T-1}\sum_{t=1}^{T-1} log(P^{lm}(w_{t+1}|w_{<t+1})) - \frac{1}{T-1}\sum_{t=2}^{T} log(P^{lm}(w_{t-1}|w_{>t-1})) \quad (5)$$

Dataset	SQuAD				MARCO			
Model	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
NQG++ (Zhou et al., 2017)	-	-	-	13.29	-	-	-	-
matching strategy (Song et al., 2018)	-	-	-	13.91	-	-	-	-
Maxout Pointer (sentence) (Zhao et al., 2018)	**44.51**	**29.07**	21.06	15.82	-	-	-	16.02
answer-focused model (Sun et al., 2018)	42.10	27.52	20.14	15.36	46.59	33.46	24.57	18.73
position-aware model (Sun et al., 2018)	42.16	27.37	20.00	15.23	47.16	34.20	24.40	18.19
hybrid model (Sun et al., 2018)	43.02	28.14	20.51	15.64	48.24	35.95	25.79	19.45
Our Model								
pointer generator with features (baseline)	41.25	26.76	19.53	14.89	54.04	36.68	26.62	20.15
w/ features + language modeling	42.80	28.43	**21.08**	**16.23**	54.47	37.30	**27.31**	**20.88**
w/o features + language modeling	42.72	27.73	20.26	15.43	**54.62**	**37.37**	27.18	20.71
w/ features + 1-layer encoder	42.12	27.48	20.12	15.33	53.51	36.42	26.49	20.11

Table 1: Experimental results of our model in different settings comparing with previous methods on two datasets.

2.3 Multi-task Learning

Instead of sharing representations between two tasks (Rei, 2017) or encoding two tasks at the same level (Liu et al., 2018; Chen et al., 2018; Kendall et al., 2018), we adopt a hierarchical structure to combine the two tasks, by treating language modeling as a low-level task and pointer generator network as high-level, because language modeling is fundamental and its semantic information will benefit question generation. In details, we first feed the input sequence into the language modeling layer to get a sequence of hidden states. Then we concatenate them with the input sequence to obtain the input of the feature-rich encoder.

Finally, the loss of LM is added to the main loss to form a combined training objective:

$$E^{total} = E + \beta E^{lm} \quad (6)$$

where β is a hyper-parameter, which is used to control the relative importance of two tasks.

3 Experiments

3.1 Dataset

We conduct experiments on two reading comprehension datasets: SQuAD and MARCO, using the data shared by Zhou et al. (2017) and Sun et al. (2018), where the lexical features are extracted with Stanford CoreNLP. In details, 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.

3.2 Experiment Settings

Our vocabulary contains the most frequent 20,000 words in each training set. Word embeddings are initialized with the pre-trained 300-dimensional Glove vectors, and are allowed to be fine-tuned during training. The representations of answer position, POS tags, NER tags and word cases are randomly initialized as 32-dimensional vectors, respectively. The encoder of our baseline model consists of 2 BiLSTM layers, and the hidden size of both the encoder and decoder is set to 512.

In our joint model, grid search is used to determine β and results are shown in Figure 2. Consequently, we set the value of β to 0.6.

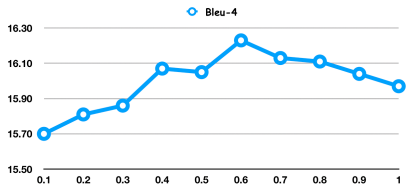

Figure 2: The impact of β on BLEU-4

We search the best-trained checkpoint base on the dev-set. In order to mitigate the fluctuation of the training procedure, we then average the nearest 5 checkpoints to obtain a single averaged model. Beam search is used with a beam size of 12.

3.3 Automatic Evaluation

Results on BLEU The experimental results on BLEU (Papineni et al., 2002) are illustrated in Table 1. Our full model (*w/ features + language modeling*) significantly outperforms previous models and achieves state-of-the-art results on both datasets, with 16.23 BLEU-4 score on SQuAD and 20.88 on MARCO respectively.

Results without Features To investigate the robustness of our model, we conduct an experiment whose input sequence only takes word embeddings and answer position, but without lexical fea-

Dataset / Model	SQuAD perplexity	SQuAD distinct-1	SQuAD distinct-2	MARCO perplexity	MARCO distinct-1	MARCO distinct-2
pointer generator with features (baseline)	41.24	9.67	39.46	17.69	16.84	45.61
w/ features + language modeling	**34.09**	**9.80**	**40.94**	15.17	**17.31**	**47.51**
w/o features + language modeling	38.70	9.73	40.89	**14.07**	17.13	46.98
w/ features + 1-layer encoder	38.40	9.56	39.71	17.61	17.10	46.87

Table 2: *Perplexity* and *distinct* of different setting models on two datasets

Dataset / Model	SQuAD Matching	SQuAD Fluency	SQuAD Relevance	MARCO Matching	MARCO Fluency	MARCO Relevance
pointer generator with features (baseline)	0.983	1.573	1.540	1.133	1.667	1.593
+ language modeling	**1.147**	**1.690**	**1.600**	**1.160**	**1.720**	**1.603**
kendall correlation coefficient	0.820	0.814	0.796	0.852	0.792	0.824

Table 3: Human evaluation results on two datasets.

tures (*w/o features + language modeling*). We can see that the auxiliary task of language modeling boosts model performance on both datasets, demonstrating that our model guarantees higher stability because it does not depend on the quality of lexical features. Therefore, our model can apply to low-resource languages where there is not adequate data for training a well-performed model for lexical features extraction.

Results with a 3-layer Encoder To validate that we gain the improvement not due to a deeper network, we replace the language modeling module with one encoder layer, that is to say, we adopt a 3-layer encoder. Comparing this model (*w/ features+1-layer encoder*) with the full model (*w/ features+language modeling*), we can see that our joint-learning model performs better than simply adding an extra encoding layer. The results on MARCO also clearly show that a deeper network does not guarantee better performance.

Perplexity and Diversity Since BLEU only measures a hard matching between references and generated text, we further adopt *perplexity* and *distinct* (Li et al., 2016) to judge the quality of generated questions. The results in Table 2 indicate that the language modeling task helps the model to generate more fluent and readable questions. Besides, the generated questions have better diversity.

3.4 Human Evaluation

For a better study on the quality of generations, we perform human evaluation. Three annotators are asked to grade the generated questions in three aspects: $matching$ indicates whether a question can be answered with the given answer; $fluency$ indicates whether a question is fluent and grammatical; $relevance$ indicates whether a question can be answered according to the given context. The rating score ranges from 0 to 2. We randomly sample 100 cases from each dataset for evaluation. Results are displayed in Table 3. The coefficient between human judges is high, validating a high quality of our annotation. The results show that by incorporating language modeling, the generated questions receive higher scores across all three metrics.

Context: Prior to the early 1960s, access to the forest's interior was highly restricted, and the forest remained basically intact.
Answer: The early 1960s
Reference: Accessing the Amazon rainforest was restricted before what era?
Baseline: When did access to the forest's interior?
Joint-model: When did access to the forest's interior become restricted?

Context: This teaching by Luther was clearly expressed in his 1525 publication on the bondage of the will, which was written in response to on free will by Desiderius Erasmus (1524).
Answer: 1525
Reference: When did Luther publish on the bondage of the will?
Baseline: In what year was the bondage of the will on the bondage of the will?
Joint-model: When was the bondage of the will published?

Table 4: Examples of generated questions by different models.

3.5 Case Study

Further, Table 4 gives two examples of the generated questions on SQuAD dataset, by the base-

line model and our joint model respectively. It is obvious that questions generated by our proposed model are more complete and grammatical.

4 Conclusion

This paper proves that equipped with language modeling as an auxiliary task, the neural model for QG can learn better representations that help the decoder to generate more accurate and fluent questions. In future work, we will adopt the auxiliary language modeling task to other neural generation systems to test its generalization ability.

Acknowledgments

We thank Weikang Li and Xin Jia for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (61773026, 61572245).

References

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

Rich Caruana. 1998. Learning to learn. chapter Multitask Learning, pages 95–133. Kluwer Academic Publishers, Norwell, MA, USA.

Ying Chen, Wenjun Hou, Xiyao Cheng, and Shoushan Li. 2018. Joint learning for emotion classification and emotion cause detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 646–651.

Kenneth Mark Colby. 1975. *Artificial Paranoia*. Elsevier Science Inc., New York, NY, USA.

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.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. 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 1342–1352.

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.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. 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 687–697.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1923–1933.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7482–7491.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119.

Lizhen Liu, Xiao Hu, Wei Song, Ruiji Fu, Ting Liu, and Guoping Hu. 2018. Neural multitask learning for simile recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1543–1553.

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 *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016.*

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.

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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. 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 2121–2130.

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.

Hai Ye and Lu Wang. 2018. Semi-supervised learning for neural keyphrase generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4142–4153.

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.