Question Generation based Product Information

Kang Xiao, Xiabing Zhou(✉), Zhongqing Wang, Xiangyu Duan and Min Zhang

Natural Language Processing Lab, School of Computer Science and Technology,
Soochow University, Suzhou, China
20184227061@stu.suda.edu.cn
{zhouxiabing,wangzq,xiangyuduan,minzhang}@suda.edu.cn

Abstract. With the continuous development of the Internet, the field of e-commerce generates many comments on products. It is of great significance for both merchants and customers to generate product-related questions by utilizing a large amount of review information of products. In order to get rid of the traditional constraints of generating models based on artificial rules and make the question generation more accurate, this paper proposes a question generation model based on product information. Compared with the existing approaches, this model can generate questions more relevant to the products, and more fluent. In particular, the model can not only avoid the problem that the vocabulary exceeds the dictionary, but also extract the vocabulary needed for question generation from the original text and the dictionary. The experimental results show that in the task of generating short text based on comments, compared with the existing neural network model, the effectiveness has been greatly improved.

Keywords: Product Information, Neural Network, Attention Replication Mechanism, Entity Recognition, Question Generation Model.

1 Introduction

With the development of the Internet era, e-commerce has become one of the most dynamic economic activities in the country. China has the world's largest e-commerce market, with about 533 million shopping users [1]. Through the analysis of commodity review information, it is helpful for more customers to understand commodity information in detail and merchants to improve product quality.

Faced with a large number of comments, it is difficult for merchants and customers to catch the key product information. Through some product Q&A information, merchants can more intuitively understand the aspects that customers concern about, and customers can more directly find what they need. However, the amount of Q&A information in reality is often much smaller than that in comments. As shown in table 1, the amount of comment information and Q&A information of 5 products is counted from an e-commerce platform. Therefore, it is a challenging task how to automatically generate commodity questions from a large amount of comment information.

Table 1. Product reviews and Q&A quantity

Item	TV	Oven	Cellphone	Electric kettle	Microwave oven
Comment number	43837	18050	33448	97843	18743
Question number	506	112	274	419	104

Automatic question generation is a research hotspot in the field of natural language processing. The traditional method which is mainly through manually set relevant rules or templates to generate questions. For example, Labutov et al. [2] put forward the question of manual text-based template construction. However, such method required lots of manpower and generated question patterns that were relatively fixed and inflexible, especially when applied to new areas, the new rules and templates still need to be defined. Recently, more and more scholars are trying to use neural network model to generate questions. For example, Serban et al. [3] proposed a sequence-to-sequence neural network based on structured data.

However, the current research mainly focuses on generating a natural question related to the text content, which is mostly based on the known dictionary, and cannot well solve the world out of vocabulary(OOV) problem. The main content of this paper is to generate questions related to product information based on comments. The characters of comments are shorter and more colloquial, and they are more prone to the new and unrecorded words. Also, generating questions must be relevant to the product. Therefore, the previous question generation model cannot well solve the above challenges, and it is extremely easy to generate inaccurate words and unsmooth sentences, as shown in the following example:

[E1] *评论信息：反应速度快，外观漂亮，屏幕也大，显示很鲜艳。*

(comment: Quick response, beautiful appearance, the screen is also large, the display is very bright)

生成问题：效果怎么样？

(generated question: what is the effect)

[E2] *评论信息：一次失败的购物体验首先手机宽度，完全处于一只手拿大，两只手拿小，没有外边框，极其容易误触。*

(comment: A failed shopping experience first of all, the width of the phone, completely in one hand to hold large, two hands to hold small, no outer border, extremely easy to miscontact.)

生成问题：手机宽外面好吗？

(generated question: the phone wide out ok?)

As can be seen from E1, when there is no word matching the product information in the given dictionary, the generated question will be inconsistent with the product content. In E2, the model based on neural network is unable to accurately divide the boundary of entities, and it is extremely easy to make mistakes in selecting words.

In order to solve the above challenges, in this paper, we propose a Question Generation based on Product Information (QGPI). In this model, the information entities related to the products are first marked, which makes the generated question more related to products. Secondly, this model uses the sequence-to-sequence model [4] based on replication mechanism. When the word is not included in the vocabulary, the original words in comment are selected, which avoids the OOV problem and makes the generated questions more smooth and flexible. The experimental results show that, the ROUGE value and BLEU value of our model both better than other models.

2 Related Works

In the past two decades, question generation methods are mainly divided into two categories: rule-based methods and neural network-based methods.

Traditional question generation is primarily rule-based or template-based. They convert the input sentence into a syntactic representation, which is then used to generate questions. Mostow et al. [5] generated self-questioning strategies for reading comprehension, which define three templates (how, what, why) to generate question. Mannem et al. [6] introduced a semantic-based system that uses syntax to help generate question. Lindberg et al. [7] created the question by using the main semantic information to build the system template. Chali and Hasan [8] used the topic model to identify the topic of sentences as heuristic rules and generate question through the entity and predicate parameter structure of sentences. Mazidi and Tarau [9] considered the frequency of sentence patterns and the consistency of semantic information transmitted by sentence patterns to produce question. However, this class of methods has some common disadvantages: dependencies and non-portability. Systems are often difficult to maintain because the rules may vary from person to person. At the same time, most systems are not easily migrated to other domains because they only have relevant rules set by proprietary domains.

In order to break through the shackle of the traditional method based on the custom rules, many scholars begin to use the neural network model to solve the question generation task. Serban et al. [3] proposed a sequence-to-sequence neural network model based on structured data (subject, relation, object) to generate simple factual question. Du et al. [10] proposed a sequence-to-sequence model of attention mechanism based on the state of encoder, and added some characteristics of words in the encoder layer to generate question. Zheng et al. [11] used a template-based method to construct the questions in key sentences, and sorted all the questions by using the multi-feature neural network model to select the top1 question. Bao et al. [12] proposed a dual antagonism network to generate cross-domain question. Different from previous research on question generation, this paper is based on product review data, which is often illogical and colloquial. People with different meanings have different expressions, and OOV problems are more likely to occur in this question generation. At the same time, question generation based on product reviews requires that the question is closely related to the product. Therefore, the previous question generation model cannot well solve the above challenges.

3 Question Generation based on Product Information

3.1 The Framework of QGPI

In this section, we introduce the QGPI, and the general framework is shown in Fig. 1. Firstly, the model annotates some entities related to product information to strengthen the correlation between generated question and products. Secondly, long short-term memory (LSTM) is used to learn the text information of comments. Thirdly, the attention mechanism is used to retain important content and identify relevant entities. Finally, the important words retained in the text are combined with the existing vocabulary by replication mechanism to make the words more accurate and the generated question statements smoother.

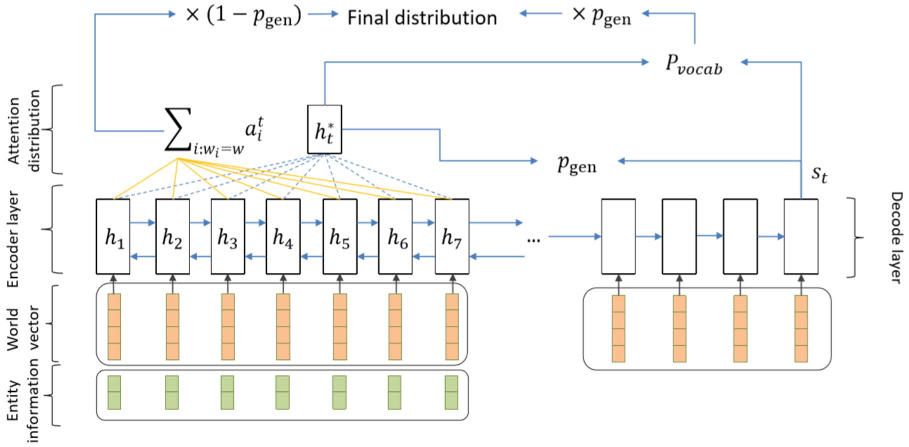

Fig. 1. Overview of the QGPI framework

3.2 Document Representation

We represent each comment d as n words$\{w_1, w_2, w_3, \ldots, w_n\}$. We transform each token w_i into its corresponding word vector x_i. For comment d, the relevant content of the comment is learned by using LSTM model [13]. On the basis of forward LSTM, backward LSTM is introduced. Each time step t in a bi-directional LSTM model produces the document representation of vector h_t by forward sequence $\overrightarrow{h_t}$ hidden layer and hidden layer of backward sequence $\overleftarrow{h_t}$:

$$\overrightarrow{h_t} = \mathrm{LSTM}(h_{t-1}, x_t) \tag{1}$$

$$\overleftarrow{h_t} = \mathrm{LSTM}(h_{t+1}, x_t) \tag{2}$$

$$h_t = W_{\vec{h}}\overrightarrow{h_t} + W_{\overleftarrow{h}}\overleftarrow{h_t} + b_t \tag{3}$$

The decode layer is a unidirectional LSTM network structure. In training, the text representation of the corresponding question is received; In testing, the state emitted by the previous layer is captured. Finally, after passing through the decoding layer, a decoding state s_t will be generated.

3.3 Product Information Labeling

In order to better generate product-related questions, the model proposed in this paper especially imparts product-related entity information into the text learning process. By labeling the entities related to product information, the generated questions are more inclined to develop questions around these entities. Experimental results show that this method is effective.

Therefore, in the process of learning comment information, it is necessary to judge whether the word is an entity and mark it. [1,0] label after the word vector when the word is an entity:

$$x_i' = contact(x_i, [1,0]) \tag{4}$$

when the word is not an entity, we add the label [0,1] after the word vector:

$$x_i' = contact(x_i, [0,1]) \tag{5}$$

where $contact$ is a connection function, whose main function is to connect two vectors head to tail. The newly generated word vector x_i' is then input into the network.

After adding the label, the spatial distance between entity and non-entity vectors will increase, which is conducive to network differentiation.

3.4 Attention Mechanism

In order to learn more accurate questions and enhance the influence of product-related content, the attention mechanism is introduced [14]. Attention mechanism combines comment information and question information to extract the important words generated by the final question. Attention mechanism uses comment information to represent h_i and question information to represent s_t to construct the weight of words in text comments:

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{attn}) \tag{6}$$

$$a^t = \mathrm{softmax}(e^t) \tag{7}$$

where v, W_h, W_s, b_{attn} are learnable parameters.

The dictionary information is added at the end of the model to fully account for the fact that the words in the generation question come not only from the comments themselves, but also from words not included in the comments. Through the weighted sum of hidden layer state generated based on the attention mechanism h_t^* , and the decode layer state s_t , the probability distribution of the generation of related question of vocabulary learning in the dictionary can be written:

$$h_t^* = \sum_i a_i^t h_i \tag{8}$$

$$P_{vocab} = \text{softmax}(V'(V[s_t, h_t^*] + b) + b') \tag{9}$$

where V, V', b and b' are learnable parameters. P_{vocab} is the probability distribution of all the words in the vocabulary.

3.5 Replication Mechanism

In this model, in order to better balance the vocabulary from the dictionary or the comment itself and avoid OOV phenomenon, for each time step t, a generation probability $p_{gen} \in [0,1]$ is added, which is obtained by the calculated h_t^*, decoding the state s_t and the input x_t of the decode layer:

$$p_{gen} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}) \tag{10}$$

where vectors $w_{h^*}^T$, w_s^T, w_x^T and scalar b_{ptr} are learnable parameters. σ is the sigmoid function.

p_{gen} is equivalent to a probability sampling, can from P_{vocab} likely to get the dictionary words, can also copy the word in the original comment. So you get a probability distribution of all words:

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i = w} a_i^t \tag{11}$$

It can be noted from equation (11) that if w is not in the vocabulary, then $P_{vocab}(w) = 0$, then the words generated in the question come from the content associated with the product from the comments, avoiding the OOV problem. In contrast, many sequence-to-sequence models are limited by predefined vocabulary, resulting in inaccurate or incorrect terms.

The final loss function of the model is shown in equation (12) - (13). At each time step t, the loss function is the negative logarithmic likelihood expression of the target vocabulary w_t^* :

$$loss_t = -\log(P(w_t^*)) \tag{12}$$

$$\text{loss} = \frac{1}{T} \sum_{t=0}^{T} loss_t \tag{13}$$

4 Experiment

4.1 Experimental Data

The data of this experiment are mainly from the comments of related products on JD, a Chinese e-commerce platform, and a series of questions about the products raised by consumers. The amount of data obtained is shown in table 2:

Table 2. Experiment data

Item	camera	cellphone	oven	pad	TV	microwave oven	juicer	pot
Comment number	62973	90983	36271	77644	58115	25433	27843	28743
Question number	1091	2834	1418	1448	1334	1120	1019	1136

For accessing to the data of this paper, we made the following treatment: World2vec was used for similarity calculation. First of all, we remove comments or questions with calculated similarity greater than 70% and then for each question, pick comments that have a similarity above 60%, at last we select the top 3 comments from similarity ranking, and splicing them together as a short text, thus forming a question with a pair of matching short text. The data form is shown in table 3.

Table 3. The data sample

Sample	Content
Short text (top 3 comments splicing)	看了评论，抱着侥幸心理，结果还是卡烫卡烫的，看网页烫，打电话三分钟就烫烫，发烫的吓人就是感觉有点烫耐用，打游戏发烫，不过有很多高科技。 (Read the comment, holding the fluke psychology, the result is still card very hot card very hot, look at the web page very hot, make a phone call three minutes very hot, very hot scary is to feel a little hot durable, play games very hot, but there are a lot of high-tech)
question	玩游戏烫不烫？ (Is it hot to play games?)

In this paper, in order to label product-related entity information more accurately, two students marked the entity at the same time, and the third student checked the difference.

4.2 Experimental Settings

- **Data Setting**: In this paper, 80% is used as training set and 20% as test data.
- **Hyper-parameters**: The size of hidden layer is 256. The dimension of word em-bed-ding is 128. The maximum time step of the encoder is 400. The maximum time step size of decoder is 100. Batch size is 16 and learning rate is 0.15.
- **Evaluation Metric**: The results were evaluated with ROUGE and BLEU values. ROUGE is usually used to measure the "similarity" between the automatically generated text and the reference text. This experiment used ROUGE to evaluate the difference between the generated question and the original question, which more reflected the semantic level. BLEU is mainly used in the field of machine translation, which requires high accuracy of translated words. This experiment is used to measure the accuracy of generating question words.

4.3 Results and Analysis

Baselines: This experiment mainly carried out five groups of comparative experiments, and the comparative experimental model is as follows

- **Seq2seq**: Sequence to sequence model based on LSTM.
- **Seq2seq+attn**: Sequential to sequential model of LSTM based on attention mechanism.
- **NQG_NER**: Du et al. [10] proposed a sequence to sequence model based on attention mechanism, adding some characteristics of words in the encode layer. This experiment is mainly to add the entity characteristics.
- **Pointer-generator**: See et al. [4] proposed a sequence-to-sequence model based on replication coverage mechanism, which can avoid OOV problems. This model has been proved feasible in the field of abstract generation.

Table 4. Automatic evaluation results of different systems by ROUGE and BLEU

Model	ROUGE			BLEU		
	1-gram	2-gram	L	value	1-gram	2-gram
Seq2seq	0.14	0.03	0.13	3.70	21.1	6.1
Seq2seq+attn	0.14	0.03	0.14	2.47	24.8	8.1.
NQG_NER	0.16	0.04	0.15	5.26	32.8	9.7
Pointer-generator	0.24	0.09	0.23	5.88	49.2	20.2.
Our model	**0.26**	**0.10**	**0.25**	**11.02**	**50.2.**	**22.2**

From ROUGE values of different models in table 4, it can be seen that the pointer-generator model and GQPI have significantly better effects than the other three sequence-based models. The main reason is that the data based on product reviews tend to be illogical and colloquial, so OOV problem can easily occur in the generation of question, which cannot be well solved by previous models based on neural network. Since both GQPI and the pointer-generator are based on replication coverage mechanism, it can select not only the corresponding words from the dictionary, but also the relevant words from the original text, thus making the sentence smoother. At the same time, the addition of product-related information in GQPI makes the generated questions revolve around the product, and performs better than pointer-generator model.

From BLEU values, it can be seen that our model is obviously superior to other model, and in the comparison with the Pointer-generator model, the BLEU value is 5 percentage points higher, which further indicates that the model in this paper is more accurate in generating.

For product questions often revolve around the entity information related to product, in addressing the review data, due to the increased the term entity information in the network, which makes the model for the division of physical boundaries and draw more accurate, raised the questions generated by the accuracy, also let the generated questions more in line with the actual situation of the product.

4.4 Result Sample Analysis

In order to better understand the effect of network, table 5 shows three examples of network generation question, from which relevant reasons can be analyzed.

Table 5. Sample of questions generated by seq2seq+attn, pointer-generator and our model.

Item	Content
Short text	屏幕除了处理器电池，其余都不错，屏幕比 7p 阴阳屏好多了还行，屏幕反应有延迟屏幕解锁比后置慢，后盖仔细看有很多凹点，处理得不好，其余有很多高科技。 (The screen is good except for the processor battery. The screen is much better than the 7p Yin and Yang screen. The back cover has a lot of concave points carefully, the treatment is not good, the rest has a lot of high-tech.)
Question	屏幕怎么样，有阴阳屏吗？ (What about the screen? Is there a Yin and Yang screen?)
Seq2seq+attn	处理怎么样？ (How about treatment?)
Pointer-generator	你们屏幕开吗？ (Is your screen on?)
Our model	屏幕怎么样？ （What about the screen?）

From the first question of network model generation, it can be seen that when OOV problem occurs, that is, "屏幕" is not in the vocabulary, the general neural network will choose the word with the highest probability from the vocabulary, and even cannot generate related words. The resulting question is very different from the standard question.

From the question generated by the pointer-generator model, it can be found that the model can select the words of the original text, but the sentence may not be smooth, mainly because the network cannot accurately identify the entities, and fail to accurately divide the boundaries of the entities.

Through comparison, it can be found that although the question generated by the model proposed in this paper cannot completely reproduce the content of the standard question, it reflects the focus of the question. It also uses the entity vocabulary accurately, and forms a smoother sentence.

5 Conclusion

This paper proposes a comment question generation model based on product information, which uses the replication mechanism of attention. When the word is not included in the vocabulary, the original words are selected to solve the OOV problem. The model is based on the short text of the comment data and the more colloquial character. In addition, the entity information of the text is added, so that the generated question is more concerned with the product itself, and the sentences are smoother. The experimental results show that our model has a better effect compared with other neural network models.

Acknowledgements. This research work has been partially supported by two NSFC grants, NO. 61702518 and NO. 61806137.

References

1. Analysis of the current market situation and forecast of the development trend of China's e-commerce industry in 2018, http://www.chyxx.com/industry/201808/666932.html, last accessed 2019/04/28.
2. Labutov, I., Basu, S., Vanderwende, L.: Deep questions without deep understanding. In: ACL, pp. 889–898 (2015).
3. Serban, I.V., Garc á-Dur án, A., Gulcehre, C., Ahn, S., Chandar, S., Courville, A., Bengio, Y.: Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. arXiv preprint arXiv:1603.06807 (2016).
4. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: ACL, pp.1073–1083 (2017).
5. Mostow, J., Chen, W.: Generating Instruction Automatically for the Reading Strategy of Self-Questioning. In AIED, pp. 465-472 (2009).
6. Mannem, P., Prasad, R., Joshi, A.: Question generation from paragraphs at UPenn: QGSTEC system description. In QG2010, pp. 84-91 (2010).
7. Lindberg, D., Popowich, F., Nesbit, J., Winne, P.: Generating natural language questions to support learning on-line. In EWNLG, pp. 105-114 (2013).
8. Chali, Y., Hasan, S. A.: Towards topic-to-question generation. Computational Linguistics 41(1), 1-20 (2015)
9. Mazidi, K., Tarau, P.: Infusing nlu into automatic question generation. In INLG, pp. 51-60 (2016).
10. Du, X., Shao, J., Cardie, C.: Learning to ask: Neural question generation for reading comprehension. In: ACL, pp. 1342-1352 (2017).
11. Zheng, H. T., Han, J., Chen, J. Y., Sangaiah, A. K.: A novel framework for Automatic Chinese Question Generation based on multi-feature neural network model. Comput. Sci. Inf. Syst. 15(3), 487-499 (2018).
12. Bao, J., Gong, Y., Duan, N., Zhou, M., Zhao, T.: Question generation with doubly adversarial nets. IEEE/ACM Trans. Audio, Speech & Language Processing 26(11), 2230-2239 (2018).
13. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks 18(5-6), 602-610 (2015).
14. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014).