Review-based Question Generation with Adaptive Instance Transfer and Augmentation[*]

Qian Yu[1,2†] **Lidong Bing**[2] **Qiong Zhang**[2] **Wai Lam**[1] **Luo Si**[2]
[1] The Chinese University of Hong Kong
[2] DAMO Academy, Alibaba Group
[1]{yuqian, wlam}@se.cuhk.edu.hk
[2]{l.bing, qz.zhang, luo.si}@alibaba-inc.com

Abstract

While online reviews of products and services become an important information source, it remains inefficient for potential consumers to exploit verbose reviews for fulfilling their information need. We propose to explore question generation as a new way of review information exploitation, namely generating questions that can be answered by the corresponding review sentences. One major challenge of this generation task is the lack of training data, i.e. explicit mapping relation between the user-posed questions and review sentences. To obtain proper training instances for the generation model, we propose an iterative learning framework with adaptive instance transfer and augmentation. To generate to the point questions about the major aspects in reviews, related features extracted in an unsupervised manner are incorporated without the burden of aspect annotation. Experiments on data from various categories of a popular E-commerce site demonstrate the effectiveness of the framework, as well as the potentials of the proposed review-based question generation task.

1 Introduction

The user-written reviews for products or service have become an important information source and there are a few research areas analyzing such data, including aspect extraction (Bing et al., 2016; Chen et al., 2013), product recommendation (Chelliah and Sarkar, 2017), and sentiment analysis (Li et al., 2018; Zhao et al., 2018a). Reviews reflect certain concerns or experiences of users on products or services, and such information is valuable for other potential consumers. However, there are few mechanisms assisting users for efficient review digestion. It is time-consuming for users to locate critical review parts that they care about, particularly in long reviews.

We propose to utilize question generation (QG) (Du et al., 2017) as a new means to overcome this problem. Specifically, given a review sentence, the generated question is expected to ask about the concerned aspect of this product, from the perspective of the review writer. Such question can be regarded as a reading anchor of the review sentence, and it is easier to view and conceive due to its concise form. As an example, the review for a battery case product in Table 1 is too long to find sentences that can answer a user question such as “How long will the battery last?”. Given the generated questions in the right column, it would be much easier to find out the helpful part of the review. Recently, as a topic attracting significant research attention, question generation is regarded as a dual task of reading comprehension in most works, namely generating a question from a sentence with a fixed text segment in the sentence designated as the answer (Duan et al., 2017; Sun et al., 2018).

Two unique characteristics of our review-based question generation task differentiate it from the previous question generation works. First, there is no review-question pairs available for training, thus a simple Seq2Seq-based question generation model for learning the mapping from the input (i.e. review) to the output (i.e. question) cannot be applied. Even though we can easily obtain large volumes of user-posed review sets and question sets, they are just separate datasets and cannot provide any supervision of input-output mapping (i.e. review-question pair). The second one is that different from the traditional question generation, the generated question from a review sentence will not simply take a fixed text segment in the review as its

[*] The work described in this paper is partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14204418).

[†] The work was done when Qian Yu was an intern at Alibaba.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 280–290
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

Review	Question
It doesn't heat up like most of the other ones, and I was completely fascinated by the ultra light and sleek design for the case. Before I was using the Mophie case but I couldn't wear it often because it was like having a hot brick in your pocket, hence I had to always leave it at home. On the contrary, with PowerBear, I never take it off because I can't even tell the difference. Also it is build in a super STRONG manner and even though I dropped my phone a few times, its shock resistant technology won't let a single thing happen to the case or the phone. The PowerBear case became an extension to my phone that I never have to take off because when I charge it at night, it charges both my phone and the case. I have battery life for more than two days for normal use, i.e. not power-consuming gaming.	Does this make the phone warm during charging? Have any of you that own this had a Mophie? Does this give protection to the phone? Can this charge the phone and the extra battery at the same time? How many days it can last?

Table 1: A product review and the example questions.

answer. The reason is that some reviews describing user experiences are highly context-sensitive. For the example in Table 1, for the review "I have battery life for more than two days for normal use, i.e. not power-consuming gaming." and its corresponding example question "How many days it can last?", obviously the text segment "more than two days" is a less precise answer, while the whole review sentence is much more informative. In some other case, even such less precise answer span cannot be extracted from the review sentence, e.g. for the example question "Does this give protection to the phone?" and the review sentence "Also it is ... even though I dropped my phone ..., its shock resistant technology won't let a single thing happen to the case or the phone.". Of course here, a simple "Yes" or "No" answer does not make much sense as well, while the whole review sentence is a vivid and informative answer.

The above two unique characteristics raise two challenges for our task. The first challenge, namely lacking review-question pairs as training instances, appears to be intractable, particularly given that the current end-to-end models are very data-hungry. One instant idea is to utilize user-posed (question, answer) pairs as substitute for training. However, several instance-related defects hinder the learned generation model from being competent for the review-based question generation. Some answers are very short, e.g. "more than two days", therefore, without necessary context, they are not helpful to generate good questions. The second challenge, namely the issue that some verbose answers contain irrelevant content especially for subjective questions. To handle this challenge, we propose a learning framework with adaptive instance transfer and augmentation.

Firstly, a pre-trained generation model based on user-posed answer-question pairs is utilized as an initial question *generator*. A *ranker* is designed to work together with the *generator* to improve the training instance set by distilling it via removing unsuitable answer-question pairs to avoid "negative transfer" (Pan and Yang, 2009), and augmenting (Kobayashi, 2018) it by adding suitable review-question pairs. For selecting suitable reviews for question generation, the *ranker* considers two factors: the major aspects in a review and the review's suitability for question generation. The two factors are captured via a reconstruction objective and a reinforcement objective with reward given by the *generator*. Thus, the *ranker* and the *generator* are iteratively enhanced, and the adaptively transferred answer-question pairs and the augmented review-question pairs gradually relieve the data lacking problem.

In accordance with the second characteristic of our task, it is plausible to regard a review sentence or clause as the answer to the corresponding question originated from it. Such treatment brings in the second challenge: how can we guarantee that the generated question concentrates on the critical aspect mentioned by the review sentence? For example, a question like "How was the experience for gaming?" is not a favourable generation for "I have battery life for more than two days for normal use, i.e. not power-consuming gaming.". To solve this problem, we incorporate aspect-based feature discovering in the *ranker*, and then we integrate the aspect features and an aspect pointer network in the *generator*. The incorporation of such aspect-related features and structures helps the *generator* to focus more on critical product aspects, other than the less important parts, which is complied with the real user-posed questions.

To sum up, our main contributions are threefold. (1) A new practical task, namely question generation from reviews without annotated instance, is proposed and it has good potential for multiple applications. (2) A novel adaptive instance transfer and augmentation framework is proposed for handling the data lacking challenge in the task. (3)

Review-based question generation is conducted on E-commerce data of various product categories.

2 Related Work

Question generation (QG) is an emerging research topic due to its wide application scenarios such as education (Wang et al., 2018), goal-oriented dialogue (Lee et al., 2018), and question answering (Duan et al., 2017). The preliminary neural QG models (Du et al., 2017; Zhou et al., 2017; Du and Cardie, 2017) outperform the rule-based methods relying on hand-craft features, and thereafter various models have been proposed to further improve the performance via incorporating question type (Dong et al., 2018), answer position (Sun et al., 2018), long passage modeling (Zhao et al., 2018b), question difficulty (Gao et al., 2019), and to the point context (Li et al., 2019). Some works try to find the possible answer text spans for facilitating the learning (Wang et al., 2019). Question generation models can be combined with its dual task, i.e., reading comprehension or question answering with various motivations, such as improving auxiliary task performance (Duan et al., 2017; Yang et al., 2017; Golub et al., 2017), collaborating QA and QG model (Tang et al., 2018, 2017), and unified learning (Xiao et al., 2018).

Although question generation has been applied on other datasets, e.g., Wikipedia (Du and Cardie, 2018), most of the existing QG works treat it as a dual task of reading comprehension (Yu et al., 2018; Cui et al., 2017), namely generating a question from a piece of text where a certain text span is marked as answer, in spite of several exceptions where only sentences without answer spans are used for generating questions (Du et al., 2017; Chali and Baghaee, 2018). Such generation setting is not suitable for reviews due to the lack of (question, review) pairs and improper assumption of text span answer as aforementioned. There are works training the question generation model with the user-written QA pairs in E-commerce sites (Hu et al., 2018; Chali and Baghaee, 2018), but the practicality is limited since the questions are only generated from answers instead of reviews.

Transfer learning (Pan and Yang, 2009; Tan et al., 2017; Li et al., 2020) refers to a broad scope of methods that exploit knowledge across domains for handling tasks in the target domain. A few terms are used for describing specific methods in this learning paradigm, e.g., self-taught learning (Raina et al., 2007), domain adaptation (Long et al., 2017), etc. Based on "what to transfer", transfer learning is categorized into four groups (Pan and Yang, 2009), namely instance transfer, feature representation transfer, parameter transfer, and relational knowledge transfer. Our learning framework can be regarded as a case of instance transfer with iterative instance adaptation and augmentation.

3 The Proposed AITA Framework

For handling the aforementioned issues, we propose an Adaptive Instance Transfer and Augmentation (AITA) framework as shown in Figure 1. Since the review-related processing is always sentence-based, we use "review" for short to refer to review sentence in this paper. Its two components, namely *ranker* and *generator*, are learned iteratively. Initially, AITA simply transfers all available (question, answer) pairs and trains a *generator*. Then it will iteratively enhance the *generator* with the help of the *ranker*. The *ranker* takes a (question, answer) pair and a review as its input and calculates a ranking score s. Thus, it can rank all reviews for a given QA pair. The ranking objective incorporates the reward provided by the *generator*, which helps find out those suitable reviews to form (review, question) pairs for training (i.e. augmenting the training data). Meanwhile, the reward from the generator also helps remove unsuitable QA pairs for training, so that it makes the transfer more adaptive. Note that the *ranker* also learns to model two hidden aspect related variables for the review, which are helpful for the *generator* to ask about the major aspects in review. Such an iterative instance manipulation procedure gradually transfers and augments the training set for handling review-based question generation.

3.1 Review Ranker for Data Augmentation

There are two pieces of input text for *ranker*. The first one is the concatenation of a (question, answer) pair qa and the second one is a review sentence r. qa and r are associated with the same product. Since the *ranker* is responsible for instance augmentation that provides (question, review) pairs, it is trained to learn a score $s(qa, r)$ which can be used to return suitable r's for a given qa.

Ranking with Partially Shared Encoders. The input qa and r are encoded with two Transformer encoders with the same structure and partially shared parameters, to leverage the advantage of

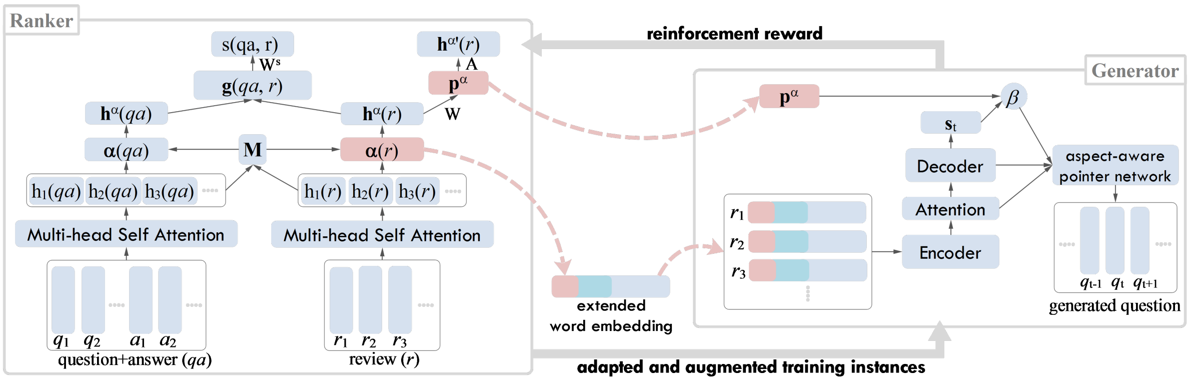

Figure 1: AITA framework. **M** is the shared parameter matrix for QA and review.

multi-head self attention on modeling word associations without considering term position. An input (qa or r) is written as a matrix $\mathrm{E} = [e_1^T, ..., e_n^T]^T$, where e is a word embedding and n is the text length. The number of heads in the multi-head self-attention is denoted as m, and the output of the j-th head is written as:

$$\mathrm{Q}^j, \mathrm{K}^j, \mathrm{V}^j = \mathrm{EW}_Q^j, \mathrm{EW}_K^j, \mathrm{EW}_V^j \tag{1}$$

$$\mathrm{head}^j(\mathrm{E}) = \mathrm{softmax}(\frac{\mathrm{Q}^j \mathrm{K}^{jT}}{\sqrt{d}})\mathrm{V}^j \tag{2}$$

where d is the dimension of word embedding. The outputs of different heads are concatenated and the encoding for the i-th word is written as $\mathbf{h}_i = [\mathrm{head}_i^1; ...; \mathrm{head}_i^m]$.

To obtain the sentence representation considering the complete semantics, we apply a global attention layer on the output of the Transformer encoder:

$$\mathbf{h}^\alpha = \sum_{i=1}^{n} \alpha_i \mathbf{h}_i \tag{3}$$

where the attention weight $\alpha_i = \exp(\mathbf{h}_i \cdot \mathrm{M} \cdot \bar{\mathbf{h}})/Z_\alpha$, Z_α is the normalization, and $\bar{\mathbf{h}} = \sum \mathbf{h}_i/n$. The parameter matrix M is shared by encoders for both qa and r for capturing the common attention features across them.

After encoding qa and r as $\mathbf{h}^\alpha(qa)$ and $\mathbf{h}^\alpha(qa)$, a vector $\mathbf{g}(qa, r)$ is assigned with the concatenation of $\mathbf{h}^\alpha(qa)$, $\mathbf{h}^\alpha(qa)$ and their difference

$$\mathbf{g}(qa, r) = [\ \mathbf{h}^\alpha(qa), \mathbf{h}^\alpha(r), |\mathbf{h}^\alpha(qa) - \mathbf{h}^\alpha(r)|\ ]$$

The review ranking score $s(qa, r)$ is calculated as:

$$s(qa, r) = \sigma(\mathrm{W}^s \mathbf{g}(qa, r) + \mathbf{b}^s) \tag{4}$$

where σ is sigmoid function.

Reinforcement Objective for Ranker Learning. To learn an appropriate $s(qa, r)$, we encounter a major challenge, namely lacking ground truth labels for (question, review). Our solution takes the *generator* in our framework as an agent that can provide reward for guiding the learning of *ranker*. The *generator* is initially trained with (question, answer) data, and is gradually updated with adapted and augmented training instances, so that the rewards from the *generator* can reflect the ability of review for generating the corresponding question.

Specifically, we propose a reinforcement objective that makes use of the reward from the *generator*, denoted as $\mathrm{reward}_G(r, q)$. For each pair of question and review, we take the normalized $\log \mathrm{ppl}(q|r)$ in the *generator* as reward:

$$\mathrm{reward}_G(r, q) = \frac{\log \mathrm{ppl}(q|r)}{\sum_{r^* \in R_{qa}} \log \mathrm{ppl}(q|r^*)} \tag{5}$$

where R_{qa} is the reviews under the same product as qa, and $\log \mathrm{ppl}(q|r)$ is the log perplexity of generating a question q from a review r:

$$\log \mathrm{ppl}(q|r) = -\frac{1}{|q|} \sum_{t \in [1, |q|]} p_G(q_t | r, q_1 ... q_{t-1})$$

The reinforcement objective for the *ranker* is to maximize the average reward for all the reviews given a question. The sampling probabilities for reviews are obtained via normalized ranking score, namely $p(r|qa) = s(qa, r)/\mathrm{Z}_{qa}$, where $\mathrm{Z}_{qa} = \sum_{r* \in \mathrm{R}_{qa}} s(qa, r*)$. The loss function is:

$$L^g(qa, r) = \mathrm{E}_{r \sim p(r|qa)} \mathrm{reward}_G(r, q) \tag{6}$$

The gradient calculation for the above objective is an intractable problem. As an approximated method which performs well in the iterative algorithm, the normalization term Z_{qa} is fixed during the calculation of the policy gradient:

$$\Delta L^{g}(qa,r) = \sum_{r} \Delta s(qa,r)\text{reward}_{G}(r,q)/\text{Z}_{qa}$$

Regularization with Unsupervised Aspect Extraction. Product aspects usually play a major role in all of product questions, answers and reviews, since they are the discussion focus of such text content. Thus, such aspects can act as connections in modeling input pairs of qa and r via the partially shared structure. To help the semantic vector $\mathbf{h}^{\alpha}$ in Eqn 3 capture salient aspects of reviews, an autoencoder module is connected to the encoding layer for reconstructing $\mathbf{h}^{\alpha}$. Together with the matrix M, the autoencoder can be used to extract salient aspects from reviews. Note that this combined structure is similar to the ABAE model (He et al., 2017), which has been shown effective for unsupervised aspect extraction. Compared with supervised aspect detection methods, such a unsupervised module avoid the burden of aspect annotation for different product categories, and our experiments demonstrate that regularization based on this module is effective.

Specifically, $\mathbf{h}^{\alpha}$ is mapped to an aspect distribution $\mathbf{p}^{\alpha}$ and then reconstructed:

$$\mathbf{p}^{\alpha} = \text{softmax}(\text{W}^{p} \cdot \mathbf{h}^{\alpha} + \mathbf{b}^{p}) \tag{7}$$

$$\mathbf{h}^{\alpha\prime} = \mathbf{p}^{\alpha} \cdot \text{A} \tag{8}$$

where each dimension in $\boldsymbol{p}^{\alpha}$ stands for the probability that the review contains the corresponding aspect, and $\boldsymbol{h}^{\alpha\prime}$ is the reconstruction of review representation, and A is a learnable parameter matrix. Note that we define "aspects" as implicit aspect categories, namely clusters of associated attributes of product, which is commonly used in unsupervised aspect extraction (Wang et al., 2015; He et al., 2017). The reconstruction objective is written as:

$$L^{\alpha}(qa,r) = [\mathbf{h}^{\alpha}(r) - \mathbf{h}^{\alpha\prime}(r)]^{2} \;/\; 2. \tag{9}$$

Only the reconstruction of review representations is considered since we focus on discovering aspects in reviews.[1] In this way, the aspect-based reconstruction will force $\mathbf{h}^{\alpha}$ to focus on salient aspects that facilitate the reconstruction. The final loss function of the *ranker* is regularized to:

$$L(qa,r) = L^{g}(qa,r) - \lambda L^{\alpha}(qa,r) \tag{10}$$

where λ is a hyper-parameter.

3.2 Question Generator in Transfer Learning

We adapt the Seq2Seq model for the aspect-focused generation model, which is updated gradually via the transferred and augmented instances. With the help of aspect-based variables learned in *ranker*, the *generator* can generate questions reflecting the major aspect in the review.

Aspect-enhanced Encoding. To emphasize the words related to salient aspects, the attention weight α_i obtained in the *ranker* is incorporated into the word embedding. Given an input review sentence, we obtain the extended word embedding $\tilde{\mathbf{e}}_i$ at position i:

$$\tilde{\mathbf{e}}_i = [\mathbf{e}_i, \mathbf{e}_i^{POS}, \mathbf{e}_i^{NER}, \alpha_i] \tag{11}$$

where $\mathbf{e}_i$ is the pre-trained word embedding, $\mathbf{e}_i^{POS}$ is the one-hot POS tag of i-th word, $\mathbf{e}_i^{NER}$ is a BIO feature for indicating whether the i-th word is a named entity, and α_i indicates the aspect-based weight for the i-th word. Bi-LSTM is adopted as the basic encoder of *generator*, encoding the i-th word as the concatenation of hidden states with both directions: $\mathbf{h}_i^{g} = [\overrightarrow{\mathbf{h}}_i, \overleftarrow{\mathbf{h}}_i]$.

Decoding with Aspect-aware Pointer Network. Pointer network, i.e., copy mechanism, can significantly improve the performance of text generation. In our task, in addition to the word-level hidden state in the decoder, the overall aspect distribution of the review can also provide clues for how likely the *generator* should copy corresponding review aspect words into the generated question.

The question is generated with an LSTM decoder. The word probability for the current time step is formulated as:

$$p_0(q_t) = \text{softmax}(\text{W}_2\tau + \mathbf{b}_2)$$

and related variables are calculated as:

$$\tau = \sigma(\text{W}_1[\mathbf{s}_t, \mathbf{c}_t] + \mathbf{b}_1)\ , \quad \mathbf{s}_t = \text{LSTM}(y_t, \mathbf{s}_{t-1})\ ,$$
$$\mathbf{c}_t = \sum_j \mathbf{z}_{tj}\mathbf{h}_j^{g}\ , \qquad \mathbf{z}_{tj} = \text{softmax}(\mathbf{h}_j^{g}\text{W}_h\mathbf{s}_t)$$

where $\mathbf{s}_t$ is the hidden state for the t-th word in question and $\mathbf{c}_t$ is the context encoding based on attention weight $\mathbf{z}_{tj}$.

In the pointer network, for a particular position t in the generated text, the word may be copied from a distribution based on the attention weight $\mathbf{z}_t$=$\{z_{tj}\}$, where the copy probability is assigned according to the current hidden state $\mathbf{s}_t$. We also

[1] We simplified the objective in AEAB model by eliminating the additional regularization term which is not necessary when combining $L^{\alpha}(qa,r)$ and $L^{g}(qa,r)$.

```
Data: QA set $\mathbf{S}_{qa}$={$(q,a)$}; review set $\mathbf{S}_r$={$r$}; $\mu$
Result: $\mathbf{S}$; generator trained with $\mathbf{S}$
Prepare pairs of $(qa, r)$ under each product
Initialize the training set $\mathbf{S} = \mathbf{S}_{qa}$
For each epoch Do
    1. Train generator with $\mathbf{S}$.
    2. Prepare the reward$_G(qa, r)$ as generator reward for each pair of $(qa, r)$
       (each answer $a$ in $qa$ pairs is regarded as a review for $q$).
    3. Adapt $\mathbf{S}$ via removing $\mu$ instances with low reward.
    4. Train ranker according to the objective in Eqn 10.
    5. Augment $\mathbf{S}$ via adding $\mu$ pairs of instances, which are $(q, r)$ pairs with top
       $s(qa, r)$ in ranker.
    6. Collect $\boldsymbol{\alpha}$ and $\boldsymbol{p}^{\alpha}$ for instances in $\mathbf{S}$ from ranker.
End
```

Algorithm 1: Learning algorithm of AITA.

consider the influence of the aspect distribution $\mathbf{p}^{\alpha}$ in the copy probability β for interpolation:

$$\beta = \sigma(\mathbf{p}^{\alpha}\mathrm{W}_c\mathbf{s}_t + \mathbf{b}_c) \tag{12}$$

The incorporation of $\mathbf{p}^{\alpha}$ helps the pointer network to consider the overall aspect distribution of context in addition to the semantics in the current position for copying words. Finally, the t-th word is generated from the mixture of the two distributions:

$$p(q_t) = (1-\beta)\cdot p_0(q_t) + \beta\cdot\mathbf{z}_t. \tag{13}$$

The *generator* is trained via maximizing the likelihood of the question q given the review r:

$$p(r|q) = \sum_i p(r_i|q, r_1, ..., r_{i-1}) \tag{14}$$

3.3 Iterative Learning Algorithm

The purpose of our iterative learning, as by Alg 1, is to update the *generator* gradually via the instance augmentation. The input data for the iterative learning consists of the transferred instance set of question-answer pairs $\mathbf{S}_{qa}$, an unlabeled review set $\mathbf{S}_r$, and an adaption parameter μ. When the learning is finished, two outputs are produced: the final training instances **S**, and the learned *generator*. The training set **S** for *generator* is initialized with $\mathbf{S}_{qa}$. In each iteration of the algorithm, the *generator* is trained with current **S**, and then **S** is adapted accordingly. The *ranker* is trained based on the rewards from the generation, which is used for instance augmentation in **S**. Thus, the training set **S** is updated during the iterative learning, starting from a pure (question, answer) set. Analysis on the influence of the composition of **S**, i.e., instance numbers of two types, is presented in Section 4.5.

There are two kinds of updates for the instance set **S**: (1) adaption via removing (q, a) pairs with low *generator* reward, in order to avoid "negative transfer"; (2) augmentation via adding (q, r) pairs that are top ranked by *ranker*, in order to increase the proportion of suitable review-question instances in training set. The instance number hyperparameter μ for removing and adding can be set according to the scale of $\mathbf{S}_{qa}$, and more details are given in our experimental setting.

To guarantee the effective instance manipulation, two interactions exist between *generator* and *ranker*. First, aspect-related variables for reviews obtained by *ranker* are part of the *generator* input. The second interaction is that a reward from *generator* is part of the learning objective for *ranker*, in order to teach *ranker* to capture the suitable reviews for generating the corresponding question.

4 Experiments

4.1 Datasets

We exploit the user-written QA dataset collected in (Wan and McAuley, 2016) and the review set collected in (McAuley et al., 2015) as our experimental data. The two datasets are collected from $Amazon.com$ separately. We filter and merge the two datasets to obtain products whose associated QA pairs and reviews can both be found. The statistics for our datasets can be found in Table 2, where the numbers of product for several very large product categories are restricted to 5000. According to the average lengths, we can find that the whole review tend to be very long. It justified our assumption that it is not easy for users to exploit reviews, and questions with short length can be a good catalogue for viewing reviews.

To test our question generation framework, we manually labeled 100 ground truth review-question pairs for each product category. 6 volunteers are asked to select user-posed questions and the corresponding review sentences that can serve as answers. Specifically, the volunteers are given pairs

	#p	#q	#a	#r	#(s)
Auto	0.8k	5.5k	18.7k	9.4k	46.5k
Baby	1.9k	11.9k	38.7k	75.3k	450.7k
Beauty	2.5k	15.9k	53.7k	62.4k	338.6k
Phones	3.6k	23.8k	87.4k	104.5k	561.8k
Cloth	0.4k	0.30k	10.7k	6.9k	32.2k
Elec	5k	31.0k	101.2k	229.4k	1461.8k
Health	5k	32.4k	114.2k	136.9k	749.9k
Music	0.4k	2.7k	8.9k	5.2k	27.9k
Sports	5k	34.2k	120.6k	122.6k	648.5k
Tools	4.1k	29.8k	104.1k	70.7k	425.6k

	L_q	L_a	L_r	L_s
Auto	14.4	23.3	88.3	17.8
Baby	15.2	22.9	106.4	17.8
Beauty	13.1	22.0	88.6	16.3
Phones	13.2	19.2	97.0	18.1
Cloth	13.0	19.8	71.2	15.3
Elec	16.1	24.8	119.5	18.8
Health	13.0	22.5	96.0	17.5
Music	14.6	24.0	94.2	17.7
Sports	13.6	22.3	91.0	17.2
Tools	14.7	23.2	110.2	18.3

Table 2: Data statistics. #: number; p, q, a, r: product, question, answer, whole review; s: review sentence, L_q, L_a, L_r, L_s are their average lengths.

of question and review, and only consider the relevance between question and review. The answer to the question is also accessible but it is only used for helping annotators to understand the question. All labeled pairs are validated by two experienced annotators with good understanding for the consumer information need in E-commerce.

.

The labeled instances are removed from the training set.

4.2 Experimental Settings

For each product category, we train the AITA framework and use the learned *generator* for testing. The fixed 300 dimension GloVe word embeddings (Pennington et al., 2014) are used as the basic word vectors. For all text including question, answer and review, we utilize StanfordNLP for tokenizing, lower casing, and linguistic features extraction, e.g., NER & POS for the encoder in *generator*. In *ranker*, the dimension of aspect distribution is set to 20 and the λ in the final loss function in Eqn 10 is set to 0.8. In the multi-head self-attention, the head number is set to 3 and the dimension for Q, K, V is 300. The dimensions of matrices can be set accordingly. The hidden dimension in *generator* is set to 200. In the iterative learning algorithm, we set the epoch number to 10 and the updating instance number μ to $0.05 \times |\mathbf{S}_{qa}|$. In testing, given a review r as input for *generator*, the additional input variables $\boldsymbol{\alpha}(r)$ and $\mathbf{p}^{\alpha}(r)$ are obtained via the review encoder (Eqn 3) and aspect extraction (Eqn 8), which are question-independent.

For testing the effectiveness of our learning framework and the incorporation of aspect, we compare our method with the following models: $\mathbf{G}_a$ (Du et al., 2017): A sentence-based Seq2Seq generation model trained with user-written answer-question pairs. $\mathbf{G}_a^{PN}$ (Wang et al., 2018): A pointer network is incorporated in the Seq2Seq decoding to decide whether to copy word from the context or select from vocabulary. $\mathbf{G}_{ar}^{PN}$: Review data is incorporated via a retrieval-based method. Specifically, the most relevant review sentence for each question is retrieved via BM25 method, and such review-question pairs are added into the training set. $\mathbf{G}_a^{PN}$+aspect (Hu et al., 2018): Aspect is exploited in this model. We trained the aspect module in our framework, i.e. only using the reconstruction objective to obtain an aspect feature extractor from reviews. Then the aspect features and distributions can be used in the same way as in our method. **AITA** refers to our proposed framework. **AITA**-aspect: All the extracted aspect-related features are removed from **AITA** as an ablation for evaluating the effectiveness of the unsupervised module for aspect. For every product category, we run each model for 3 times and report the average performance with four evaluation metrics, including BLEU1 (B1), BLEU4 (B4), METEOR (MET) and ROUGE-L (R$_L$).

4.3 Evaluation of Question Generation

The results are demonstrated in Table 3. **AITA** achieves the best performance on all product categories regarding different evaluation metrics. The significant improvements over other models demonstrate that our instance transfer and augmentation method can indeed reduce inappropriate answer-question pairs and provide helpful review-question pairs for the *generator*. The performance of $\mathbf{G}_a$ is very poor due to the missing of attention mechanism. Both $\mathbf{G}_a^{PN}$ and $\mathbf{G}_a^{PN}$+aspect have worse performance than ours, even though some product categories have large volume of QA pairs (>100k), e.g., Electronics, Tools, etc. This indicates that the answer-question instances are not capable of learning a review-based question generator because of the different characteristics between the answer set and review set. $\mathbf{G}_{ar}^{PN}$ performs much worse than $\mathbf{G}_a^{PN}$, which proves that a simple retrieval method

	BLEU1	BLEU4	METEOR	ROUGE-L	BLEU1	BLEU4	METEOR	ROUGE-L
	Automative				Baby			
G_a	0.103	0.047	0.062	0.089	0.104	0.055	0.065	0.068
G_a^{PN}	0.162	0.090	0.091	0.140	0.153	0.088	0.087	0.195
G_{ar}^{PN}	0.147	0.082	0.078	0.118	0.133	0.060	0.068	0.102
G_a^{PN}+aspect	0.165	0.090	0.093	0.140	0.157	0.088	0.091	0.203
AITA-aspect	0.179	0.094	0.094	0.146	0.157	**0.089**	0.092	214
AITA	**0.184**	**0.097**	**0.099**	**0.148**	**0.167**	**0.089**	**0.094**	**0.221**
	Beauty				Cell Phone			
G_a	0.133	0.088	0.118	0.218	0.203	0.125	0.130	0.104
G_a^{PN}	0.235	0.122	0.128	0.257	0.250	0.122	0.150	0.217
G_{ar}^{PN}	0.194	0.098	0.119	0.205	0.215	0.117	0.136	0.141
G_a^{PN}+aspect	0.240	0.122	0.132	0.257	0.251	0.134	0.154	0.223
AITA-aspect	0.240	0.127	0.132	0.257	0.261	0.139	0.184	0.230
AITA	**0.249**	**0.129**	**0.136**	**0.259**	**0.267**	**0.142**	**0.193**	**0.244**
	Clothing & Jewelry				Electronics			
G_a	0.224	0.093	0.091	0.178	0.099	0.048	0.107	0.144
G_a^{PN}	0.283	0.134	0.118	0.227	0.124	0.069	**0.131**	0.171
G_{ar}^{PN}	0.258	0.110	0.101	0.198	0.100	0.053	0.121	0.156
G_a^{PN}+aspect	0.298	0.139	0.125	0.241	0.120	0.069	0.126	0.171
AITA-aspect	0.306	0.152	0.138	0.246	0.125	0.069	**0.131**	0.174
AITA	**0.316**	**0.157**	**0.145**	**0.263**	**0.127**	**0.073**	**0.131**	**0.175**
	Health				Musical Instruments			
G_a	0.114	0.062	0.091	0.095	0.088	0.054	0.096	0.091
G_a^{PN}	0.130	0.080	0.089	0.108	0.114	0.110	0.121	0.119
G_{ar}^{PN}	0.124	0.069	0.086	0.104	0.090	0.072	0.106	0.103
G_a^{PN}+aspect	0.133	0.100	0.123	0.175	0.118	0.110	0.130	0.192
AITA-aspect	0.137	0.100	0.121	0.179	0.124	0.110	0.136	0.201
AITA	**0.142**	**0.109**	**0.132**	**0.194**	**0.129**	**0.112**	**0.141**	**0.205**
	Sports & Outdoors				Tools			
G_a	0.079	0.046	0.042	0.064	0.098	0.059	0.093	0.105
G_a^{PN}	0.091	0.052	0.079	**0.102**	0.107	0.077	0.112	0.135
G_{ar}^{PN}	0.087	0.050	0.071	0.083	0.100	0.072	0.103	0.119
G_a^{PN}+aspect	0.091	0.052	0.079	**0.102**	0.110	0.079	0.110	0.136
AITA-aspect	0.094	0.052	0.080	0.102	0.112	0.079	0.116	0.142
AITA	**0.097**	**0.057**	**0.083**	**0.102**	**0.117**	**0.083**	**0.120**	**0.149**

Table 3: Overall performance on question generation.

is not effective for merging the instances related to reviews and answers. **AITA** adapts and augments the QA set to select suitable review-question pairs considering both aspect and generation suitability, resulting in a better *generator*. In addition, effectiveness of aspect feature and aspect pointer network can be illustrated via the slight but stable improvement of $\mathbf{G}_a^{PN}$+aspect over $\mathbf{G}_a^{PN}$ and the performance drop of **AITA**-aspect on all the categories. This proves that even without precise aspect annotation, our unsupervised aspect-based regularization is helpful for improving generation.

Clothing & Jewelry			
	Relevance	*Aspect*	*Fluency*
G_a^{PN}	0.58	0.62	2.58
G_{ar}^{PN}	0.47	0.58	2.29
G_a^{PN}+aspect	0.66	0.72	2.76
AITA	**0.80**	**0.80**	**2.86**
Cell Phone			
	Relevance	*Aspect*	*Fluency*
G_a^{PN}	0.42	0.55	2.79
G_{ar}^{PN}	0.35	0.41	2.44
G_a^{PN}+aspect	0.58	0.63	2.83
AITA	**0.72**	**0.72**	**2.90**

Table 4: Performance of human evaluation.

4.4 Human Evaluation and Case Study

We conduct human evaluation on two product categories to study the quality of the generated questions. Two binary metrics *Relevance* and *Aspect* are used to indicate whether a question can be answered by the review and whether they share the same or related product aspect. The third metric, *Fluency* with the value set $\{1, 2, 3\}$, is adopted for judging the question fluency. 1 means not fluent and 3 means very fluent. We selected 50 generated questions from each model and asked 4 volunteers

The entire length of the watch is 9 inches, but the effective length from the last hole to clasp is about 8 inches.

- $\mathbf{G}_a^{PN}$: What is the difference between gear 2 neo and this watch?
- $\mathbf{G}_a^{PN}$+aspect: How is the length?
- **AITA**: What is the dimension in mm?

If you have a huge wrist this watch mayn't look good nor fit you well.

- $\mathbf{G}_a^{PN}$: What is the wrist size?
- $\mathbf{G}_a^{PN}$+aspect: How does it fit?
- **AITA**: Will it fit my huge hand?

The stainless steel case back can be pried off from the 12 o'clock position (from the back), and the battery CAN be replaced.

- $\mathbf{G}_a^{PN}$: Is the material good quality and not easy to tore?
- $\mathbf{G}_a^{PN}$+aspect: Can the lid be removed?
- **AITA**: Can you tell me how to replace the battery?

The watch has a Japanese Miyota movement inside, and has a Japanese Sony 626sw battery which requires you to loosen a very small flat head screw and slide a little metal arm out of the way to remove the battery.

- $\mathbf{G}_a^{PN}$: What is the battery life on this watch?
- $\mathbf{G}_a^{PN}$+aspect: Can I remove the battery?
- **AITA**: Can I remove the battery?

Table 5: Case study of generated questions.

for evaluation. The average scores are reported in Table 4, which shows that our framework achieves the best performance regarding all the metrics, especially for *Relevance*, showing that our AITA can help generate more accurate questions based on reviews and thus facilitates exploiting reviews. Due to the incorporation of implicit aspect information, both **AITA** and $\mathbf{G}_a^{PN}$+aspect significantly outperform $\mathbf{G}_a^{PN}$ regarding both *Aspect* and *Relevance*. Again, $\mathbf{G}_{ar}^{PN}$ with a simple retrieval method for augmenting training instances cannot perform well.

The blue sentences in Table 5 are from a long review talking about some important information of a wat ch, and the questions generated by different models are also given. These questions are more user-friendly and potential consumers can browse them to quickly locate the information they care about. For example, if a user wants to know more about the battery replacement, the portion before the third sentence can be skipped. According to the generated questions via three methods in the Table 5, we can find that the questions from AITA are asking about major aspects of the review sentences. $\mathbf{G}_a^{PN}$ failed to capture major aspects in the first three sentences, and the questions generated by $\mathbf{G}_a^{PN}$+aspect are not as concrete as ours, owning to the insufficient training instances.

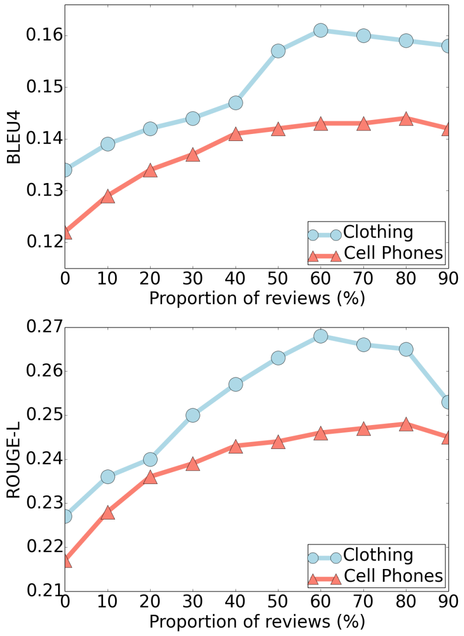

Figure 2: Analysis for proposition of instances.

4.5 Analysis on Instances Composition

The training instance set for the *generator*, i.e., **S** in Algorithm 1, is initialized with QA set and gradually adapted and augmented. Here, we investigate the effect of composition property of **S** on the *generator* performance at different epochs. As shown in Fig 2, two product categories and two metrics are illustrated, with the gradually changed training instance set **S**. The proportion of review-question (qr) instances in **S** starts with 0, and significant performance improvement can be observed while the qr proportion gradually increases. The results stay stable until the qr proportion reach 80%.

5 Conclusions

We propose a practical task of question generation from reviews, whose major challenge is the lack of training instances. An adaptive instance transfer and augmentation framework is designed for handling the task via an iterative learning algorithm. Unsupervised aspect extraction is integrated for aspect-aware question generation. Experiments on real-world E-commerce data demonstrate the effectiveness of the training instance manipulation in our framework and the potentials of the review-based question generation task.

References

Lidong Bing, Tak-Lam Wong, and Wai Lam. 2016. Unsupervised extraction of popular product attributes from e-commerce web sites by considering customer reviews. *ACM Transactions on Internet Technology*, 16:1–17.

Yllias Chali and Tina Baghaee. 2018. Automatic opinion question generation. In *INLG*, pages 152–158.

Muthusamy Chelliah and Sudeshna Sarkar. 2017. Product recommendations enhanced with reviews. In *ACM Conference on Recommender Systems*, RecSys '17, pages 398–399.

Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Exploiting domain knowledge in aspect extraction. In *EMNLP*, pages 1655–1667.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *ACL*, pages 593–602.

Xiaozheng Dong, Yu Hong, Xin Chen, Weikang Li, Min Zhang, and Qiaoming Zhu. 2018. Neural question generation with semantics of question type. In *CCF NLPCC*, pages 213–223.

Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *EMNLP*, pages 2067–2073.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. In *ACL*, pages 1907–1917.

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

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *EMNLP*, pages 866–874.

Yifan Gao, Lidong Bing, Wang Chen, Michael R. Lyu, and Irwin King. 2019. Difficulty controllable generation of reading comprehension questions. In *IJCAI*, pages 4968–4974.

David Golub, Po-Sen Huang, Xiaodong He, and Li Deng. 2017. Two-stage synthesis networks for transfer learning in machine comprehension. In *EMNLP*, pages 835–844.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *ACL*, pages 388–397.

Wenpeng Hu, Bing Liu, Jinwen Ma, Dongyan Zhao, and Rui Yan. 2018. Aspect-based question generation. In *ICLR Workshop track*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. 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 452–457.

Sang-Woo Lee, Yu-Jung Heo, and Byoung-Tak Zhang. 2018. Answerer in questioner's mind: Information theoretic approach to goal-oriented visual dialog. In *NeurIPS*, pages 2579–2589.

Jingjing Li, Yifan Gao, Lidong Bing, Irwin King, and Michael R. Lyu. 2019. Improving question generation with to the point context. In *EMNLP*, pages 3214–3224.

Juntao Li, Ruidan He, Hai Ye, Hwee Tou Ng, Lidong Bing, and Rui Yan. 2020. Unsupervised domain adaptation of a pretrained cross-lingual language model. In *IJCAI*.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *ACL*, pages 946–956.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on TKDE*, 22(10):1345–1359.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. 2007. Self-taught learning: transfer learning from unlabeled data. In *ICML*, pages 759–766. ACM.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *EMNLP*, pages 3930–3939.

Ben Tan, Yu Zhang, Sinno Jialin Pan, and Qiang Yang. 2017. Distant domain transfer learning. In *AAAI*, pages 2604–2610.

Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.

Duyu Tang, Nan Duan, Zhao Yan, Zhirui Zhang, Yibo Sun, Shujie Liu, Yuanhua Lv, and Ming Zhou. 2018. Learning to collaborate for question answering and asking. In *NAACL-HLT*, pages 1564–1574.

Mengting Wan and Julian McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. In *ICDM*, pages 489–498.

Linlin Wang, Kang Liu, Zhu Cao, Jun Zhao, and Gerard De Melo. 2015. Sentiment-aspect extraction based on restricted boltzmann machines. In *ACL*, pages 616–625.

Siyuan Wang, Zhongyu Wei, Zihao Fan, Yang Liu, and Xuanjing Huang. 2019. A multi-agent communication framework for question-worthy phrase extraction and question generation. In *AAAI*, pages 7168–7175.

Zichao Wang, Andrew S Lan, Weili Nie, Andrew E Waters, Phillip J Grimaldi, and Richard G Baraniuk. 2018. QG-Net: a data-driven question generation model for educational content. In *Annual ACM Conference on Learning at Scale*, page 7.

Han Xiao, Feng Wang, Yanjian Feng, and Jingyao Zheng. 2018. Dual ask-answer network for machine reading comprehension. *arXiv preprint arXiv:1809.01997*.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-supervised qa with generative domain-adaptive nets. In *ACL*, pages 1040–1050.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

Wei Zhao, Ziyu Guan, Long Chen, Xiaofei He, Deng Cai, Beidou Wang, and Quan Wang. 2018a. Weakly-supervised deep embedding for product review sentiment analysis. *IEEE Transactions on TKDE*, 30(1):185–197.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018b. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *EMNLP*, 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 *CCF NLPCC*, pages 662–671.