Review Response Generation in E-Commerce Platforms with External Product Information

Lujun Zhao*
Fudan University
Shanghai, China
ljzhao16@fudan.edu.cn

Kaisong Song
Alibaba Group
Hangzhou, Zhejiang, China
kaisong.sks@alibaba-inc.com

Changlong Sun
Alibaba Group
Hangzhou, Zhejiang, China
changlong.scl@taobao.com

Qi Zhang
Fudan University
Shanghai, China
qz@fudan.edu.cn

Xuanjing Huang
Fudan University
Shanghai, China
xjhuang@fudan.edu.cn

Xiaozhong Liu†
Indiana University Bloomington
Bloomington, Indiana, USA
liu237@indiana.edu

ABSTRACT

"User reviews" are becoming an essential component of e-commerce. When buyers write a negative or doubting review, ideally, the sellers need to quickly give a response to minimize the potential impact. When the number of reviews is growing at a frightening speed, there is an urgent need to build a response writing assistant for customer service providers. In order to generate high-quality responses, the algorithm needs to consume and understand the information from both the original review and the target product. The classical sequence-to-sequence (Seq2Seq) methods can hardly satisfy this requirement. In this study, we propose a novel deep neural network model based on the Seq2Seq framework for the review response generation task in e-commerce platforms, which can incorporate product information by a gated multi-source attention mechanism and a copy mechanism. Moreover, we employ a reinforcement learning technique to reduce the exposure bias problem. To evaluate the proposed model, we constructed a large-scale dataset from a popular e-commerce website, which contains product information. Empirical studies on both automatic evaluation metrics and human annotations show that the proposed model can generate informative and diverse responses, significantly outperforming state-of-the-art text generation models.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Natural language processing**; **Natural language generation**; • **Information systems** → *Electronic commerce.*

KEYWORDS

Review Response Generation, Neural Network, Sequence to Sequence Model, Gated Multi-Source Attention Mechanism, Reinforcement Learning

*Work done as an intern at Alibaba.
†Corresponding author.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313581

ACM Reference Format:
Lujun Zhao, Kaisong Song, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. 2019. Review Response Generation in E-Commerce Platforms with External Product Information. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313581

1 INTRODUCTION

Over the past few years, online shopping has experienced unimaginable growth. In e-commerce platforms, such as *Amazon*, *eBay* and *Taobao*, massive reviews from users are becoming increasingly important. From the seller viewpoint, however, reviews can be a double-edged sword. On the one hand, the more reviews, the more buys, but on the other hand, sellers have little or no control over buyer opinion. Therefore, efficient and constructive response to the target review/buyer can be critically important. If the buyers, especially those who send negative or doubting reviews, do not get a quick reply, they and other potential customers may turn to alternatives, which is unfavorable to the sellers and platforms.

Statistics[1] from a world leading e-commerce platform shows that sellers who provide high quality review responses could achieve a higher selling volume than those who rarely provide responses. However, only 2.98% of the reviews and 5.68% of the negative reviews receive responses. Furthermore, a large number of responses are universal replies which are based on the pre-defined templates/messages. Ideally, sellers should provide high-quality responses to address the personal needs from the target reviews. However, such cost can be prohibitive for most small-businesses, which inspires us to build a response writing assistant. After the algorithm automatically generates some candidate responses, the customer services could choose one and polish it, which makes the procedure more friendly and efficient.

Recently, deep neural network based methods have widely used in many natural language generation (NLG) tasks and have achieved great success, such as machine translation [3, 4, 10, 26], dialogue generation [23, 24], and text summarization [17, 22, 30]. Most of these models belong to a family of sequence-to-sequence (Seq2Seq) framework, which is composed of two parts: an encoder and a decoder. The encoder reads and encodes a source sentence into a hidden vector and the decoder outputs a target sentence based on

[1] More detail can be found in Appendices A.1.

Review: The quality is not very good and the sleeves will pill. It looks like the fabric is bad.

Seq2Seq+Attention: Thank you for your support and feedback. We will continue to work hard to provide you with better service and look forward to your next visit!

Real Response: Dear customer, thank you for choosing NAISHITU flagship store. Due to the composition of polyester, slight pilling is a normal phenomenon. It is recommended that you use fabric shaver to deal with. Looking forward to your next visit.

Figure 1: Given a review, the response generated by "Seq2Seq +Attention" model is trivial and universal. In contrast, the real response written by human contains some product information, like the *brand* (NAISHITU) and *material* (polyester).

Field	Value
Brand	NAISHITU
Material	polyester 57% cotton 43%
Process	easy care finishing
Style	casual style
Color	black
Size	185 / 2XL
...	...

Figure 2: An example of product information collected as a table.

the hidden vector. The attention mechanism [3, 16], which selectively focuses on parts of the source sentence during generation, is integrated into the Seq2Seq framework to improve the quality of long sequence generation.

Despite its enormous success, for review response generation task, however, the standard Seq2Seq+Attention model can only employ the textual reviews when generating responses. Intuitively, each review also has an associated product, and the product information can be critical and necessary to generate high-quality responses. Figure 1 illustrates an example. In this example, the given review is a negative review. Without employing the product information, the Seq2Seq+Attention model tends to generate trivial and universal responses like "*thank you for your support and feedback*", or "*we will continue to work hard to provide you with better service*" due to the high frequency of these patterns in the training data. These universal responses will give users a very bad user experience. In contrast, in real world, an experienced seller gives a response according to the corresponding product information, like the *brand* (NAISHITU) and the *material* (polyester). Without taking such product information into consideration, the algorithm can hardly generate such high-quality responses. Hence, integrating product information into the generation process has become a crucial part of the task.

In this work, the product information is collected as a factual table which contains many field-value records and an example is given in Figure 2. We follow Lebret et al. [12] to encode this table. The proposed model utilizes product information via a gated multi-source attention mechanism and a copy mechanism. The novel attention mechanism includes two attentions: review attention and product attention. For generation, we first obtain the review context vector by review attention which follows the existing attention techniques. Then, the review context vector is used to calculate the product context vector by product attention, which simulates the human response process of finding useful information in a product according to the issue mentioned in review. Finally, we obtain the final context vector through a Gated Multimodal Unit (GMU) [1], which is used to learn fusion transformations from multiple sources of information. By incorporating the copy mechanism [9], the decoder could directly copy the sub-sequences of review or product information into the response. Moreover, we employ self-critical sequence training (SCST) [21], a policy gradient reinforcement learning technique, to train the model, which reduces the exposure bias problem [20]. To evaluate the proposed model, we constructed a large-scale dataset from Taobao, which contained 100K (review, product information, response) triples. Experimental results on this dataset show the effectiveness of our method.

In summary, the contributions of this paper include the following three aspects:

(1) We introduce the task of review response generation in e-commerce platforms, which is valuable for customer service providers, but has not yet been well-studied.
(2) In order to address this problem, we propose leveraging a gated multi-source attention mechanism and a copy mechanism to integrate product information and employ policy gradient reinforcement learning in the training procedure to reduce the exposure bias problem.
(3) Experimental results based on both automatic evaluation metrics and human annotations demonstrate that the proposed model can generate informative and diverse responses, significantly outperforming state-of-the-art NLG models. We also release the large-scale review response generation dataset to enable future investigations.

2 PRELIMINARIES

In this section, we first formulate the review response generation task formally, and then introduce the field representation which will be used by our model.

Symbol	Description
$\boldsymbol{X}$	a review
$\boldsymbol{x}_i$	i-th word in review $\boldsymbol{X}$
Y	a response
$\boldsymbol{y}_i$	i-th word in response $\boldsymbol{Y}$
$\boldsymbol{T}$	a table representing product information
$\boldsymbol{t}_i$	i-th word in the sequence composed by all value
$\boldsymbol{f}_i$	the field name of i-th word $\boldsymbol{t}_i$
$\boldsymbol{p}_i^+, \boldsymbol{p}_i^-$	the position of i-th word $\boldsymbol{t}_i$
$\boldsymbol{z}_i$	the field representation of i-th word $\boldsymbol{t}_i$
$\boldsymbol{h}_i^X$	hidden state of the review encoder
$\boldsymbol{h}_i^T$	hidden state of the product information encoder
$\boldsymbol{h}_i^Y$	hidden state of the response decoder
$\boldsymbol{c}_i^X$	the review context vector
$\boldsymbol{c}_i^T$	the product information context vector
$\boldsymbol{c}_i$	the final context vector

Table 1: Commonly used notations in this paper.

Word	Field Representation
NAISHITU	(**Brand**, 1, 1)
polyester	(**Material**, 1, 4)
57%	(**Material**, 2, 3)
cotton	(**Material**, 3, 2)
...	...
casual	(**Style**, 1, 2)
style	(**Style**, 2, 1)
...	...

Figure 3: The table is represented as a sequence of words (the first column) and their corresponding field representation (the second column).

2.1 Task Definition

We first introduce our key notations used in this paper. Table 1 lists the main notation we use. Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ denote a review of length n, where $\boldsymbol{x}_i \in \mathbb{R}^{d_w}$ is the word embedding of i-th word. In the same way, let $Y = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m\}$ denote the corresponding response of length m. The product information is represented as a table $\boldsymbol{T}$, which is a combination of many field-value records. Figure 2 shows an example. In this example, we only show six fields: *brand*, *material*, *process*, *style*, *color* and *size*. Each value is a sequence of words and all values are concatenated as a long sequence $\{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_k\}$ (the first column in Figure 3) where $\boldsymbol{t}_i \in \mathbb{R}^{d_w}$ is the word embedding vector. To incorporate the field information, each word $\boldsymbol{t}_i$ also has a corresponding field representation z_i (the second column in Figure 3), which will be detailed in the next subsection. Given a review $\boldsymbol{X}$ and a table $\boldsymbol{T}$ representing the associated product information, the aim is to build a model that can generate a response $\boldsymbol{Y}$.

We formulate the response generation as a probabilistic model, which is trained to maximize the generation probability of Y conditioned on $\boldsymbol{X}$ and $\boldsymbol{T}$:

$$\begin{aligned} &\operatorname{argmax} P(Y|\boldsymbol{X}, \boldsymbol{T}) \\ =& \operatorname{argmax} P(\boldsymbol{y}_1|\boldsymbol{X}, \boldsymbol{T}) \prod_{i=2}^{m} P(\boldsymbol{y}_i|\boldsymbol{y}_{1:i-1}, \boldsymbol{X}, \boldsymbol{T}). \end{aligned} \tag{1}$$

2.2 Field Representation

As seen in Figure 2 and Figure 3, the value of each field in the product information is split into separate words and the entire table is transformed into a large sequence $\{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_k\}$. Besides the value information, the field information is also crucial, especially for the product attention. For example, an user may mention "material" in his review, but not mention "polyester". If we know that the corresponding field of "polyester" is "material", it will be easier for the attention mechanism to work well. Lebret, Grangier, and Auli [12] proposed representing value's field information by its corresponding field name and its position in the field. The field representation of a word $\boldsymbol{t}_i$ in the table is described by a {field, position} pairs:

$$z_i = \{\boldsymbol{f}_i, \boldsymbol{p}_i\}. \tag{2}$$

The pair $\{\boldsymbol{f}_i, \boldsymbol{p}_i\}$ indicates that $\boldsymbol{t}_i$ occurs in field $\boldsymbol{f}_i$ at position $\boldsymbol{p}_i$. The position $\boldsymbol{p}_i$ can be further divided as $(\boldsymbol{p}_i^+, \boldsymbol{p}_i^-)$ which represents the positions of the word $\boldsymbol{t}_i$ counted from the beginning and the end of the field respectively:

$$z_i = \{\boldsymbol{f}_i, \boldsymbol{p}_i^+, \boldsymbol{p}_i^-\}, \tag{3}$$

where $\boldsymbol{f}_i \in \mathbb{R}^{d_f}$ and $\boldsymbol{p}_i^+, \boldsymbol{p}_i^- \in \mathbb{R}^{d_p}$ are the embedding vector of field name and positions, respectively. d_f is the field name embedding size and d_p is the position embedding size. For example in Figure 2, the second field-value record is (**Material**, polyester 57% cotton 43%). The word "polyester" is the first word of field "**Material**", counted from the beginning, and the fourth word counted from the end. So the field representation of "polyester" is (**Material**, 1, 4), as shown in the second row of Figure 3.

3 THE PROPOSED MODEL

3.1 Network Architecture

In this section, we describe the network architecture used in this paper, which is illustrated in Figure 4. The proposed model mainly includes four parts: an encoder to read the review (blue), an encoder to read corresponding product information (green), a Gated Multimodal Unit (GMU) (yellow) to learn fusion transformations, and a decoder to generate the response (red).

Review Encoder

We first transform each word in a given review to a one-hot vector in the size of the vocabulary. Next, we use a simple embedding

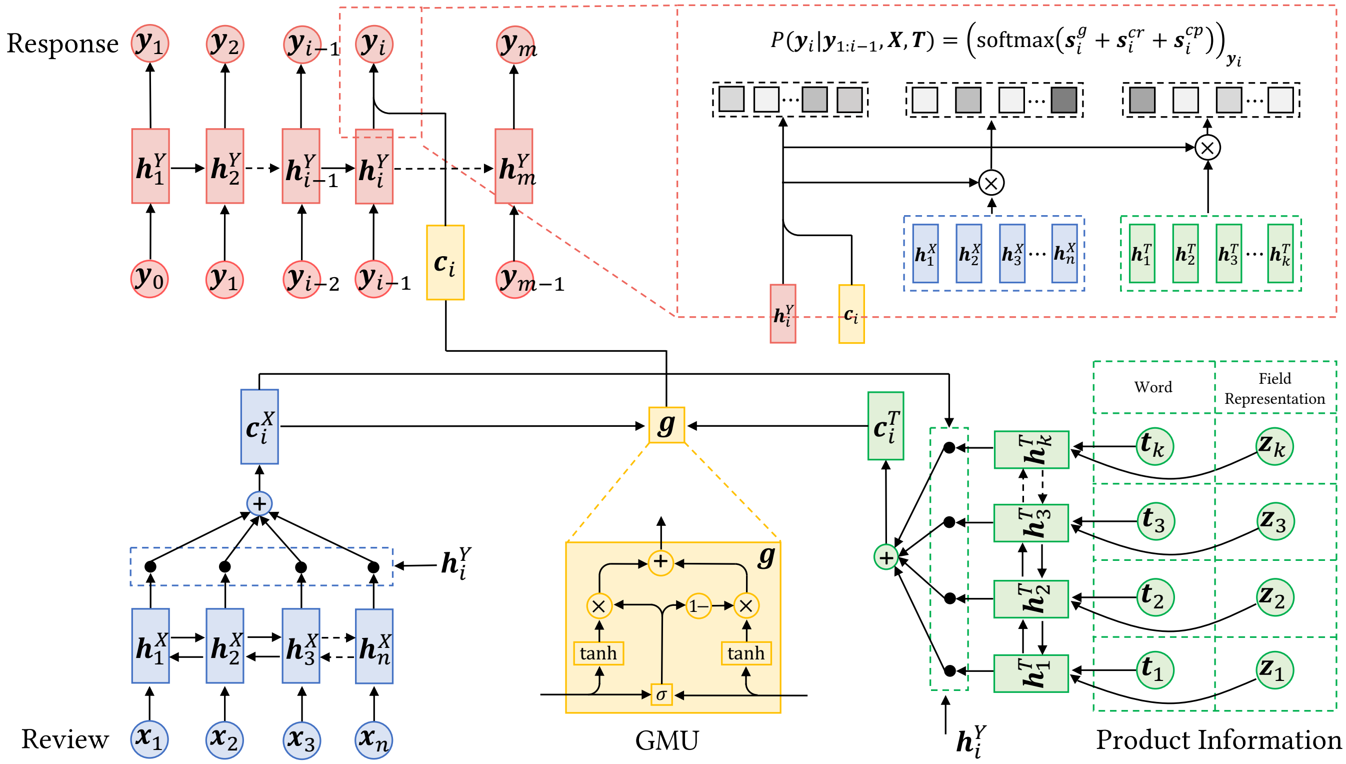

Figure 4: Architecture of our proposed model. It includes two encoder to read the review (blue) and corresponding product information (green) and a decoder to generate the response (red). A Gated Multimodal Unit (GMU) (yellow) is used to learn fusion transformations from multiple sources of information and a copy mechanism (top right corner) allows the decoder to directly copy from review or product information.

layer to encode each one-hot vector to a word vector $\boldsymbol{x}_i$. After we get the word-level review feature representation $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, the review encoder aims to encode each word $\boldsymbol{x}_i$ in the review into the hidden state $\boldsymbol{h}_i^X$. In this work, we use two-layer bidirectional recurrent neural network (BiRNN) with gated recurrent units (GRU) [4] to read the review. The bidirectional RNN can not only capture the previous contextual information but also the future contextual information. At each time step, the bidirectional RNN unit takes the word embedding $\boldsymbol{x}_i$ as input, and outputs a hidden state $\boldsymbol{h}_i^X$. Concretely, the formulas at step i are as follows:

$$\begin{aligned} \overrightarrow{\boldsymbol{h}}_i^X, \overleftarrow{\boldsymbol{h}}_i^X &= \mathrm{BiGRU}^X(\boldsymbol{x}_i, \overrightarrow{\boldsymbol{h}}_{i-1}^X, \overleftarrow{\boldsymbol{h}}_{i+1}^X; \boldsymbol{\theta}_X), \\ \boldsymbol{h}_i^X &= \overrightarrow{\boldsymbol{h}}_i^X \oplus \overleftarrow{\boldsymbol{h}}_i^X, \end{aligned} \tag{4}$$

where $\overrightarrow{\boldsymbol{h}}_i^X, \overleftarrow{\boldsymbol{h}}_i^X \in \mathbb{R}^{d_h}$ is the forward hidden state and backward hidden state of the BiGRU^X at step i, respectively; $\oplus$ is the concatenation operation and $\boldsymbol{h}_i^X$ is the final hidden state at step i. $\boldsymbol{\theta}_X$ represents all the parameters of the BiGRU^X.

Product Encoder

Besides the review encoder, we also need a product encoder to read the product information. As described above, the product information is represented as a sequence of words $\{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_k\}$ and their corresponding field representation $\{z_1, z_2, \ldots, z_k\}$. The basic structure of the product encoder is the same as that of the review encoder, i.e., two-layer bidirectional recurrent neural network (BiRNN) with gated recurrent units (GRU). The difference is that we feed the concatenation of word embedding $\boldsymbol{t}_i$ and the corresponding field representation $\boldsymbol{z}_i$ to the bidirectional RNN. Concretely, the formulas at step i are as follows:

$$\begin{aligned} \overrightarrow{\boldsymbol{h}}_i^T, \overleftarrow{\boldsymbol{h}}_i^T &= \mathrm{BiGRU}^T(\boldsymbol{t}_i \oplus z_i, \overrightarrow{\boldsymbol{h}}_{i-1}^T, \overleftarrow{\boldsymbol{h}}_{i+1}^T; \boldsymbol{\theta}_T), \\ \boldsymbol{h}_i^T &= \overrightarrow{\boldsymbol{h}}_i^T \oplus \overleftarrow{\boldsymbol{h}}_i^T, \end{aligned} \tag{5}$$

where $\overrightarrow{\boldsymbol{h}}_i^T, \overleftarrow{\boldsymbol{h}}_i^T \in \mathbb{R}^{d_h}$ is the forward hidden state and backward hidden state of the BiGRU^T at step i, respectively; $\boldsymbol{h}_i^T$ is the final hidden state. $\boldsymbol{\theta}_T$ represents all the parameters of the BiGRU^T.

Response Decoder

Once the review and product information is encoded, a decoding two-layer RNN with GRU is used to generate the response. The decoding process starts as soon as the decoder receives a starting symbol "<s>" ($\boldsymbol{y}_0$ in Figure 4) and ends when the decoder generates an ending symbol "<e>". On each step i, the decoder receives the word embedding of the previous word and outputs the decoder hidden state $\boldsymbol{h}_i^Y$:

$$\boldsymbol{h}_i^Y = \mathrm{GRU}^Y(\boldsymbol{y}_{i-1}, \boldsymbol{h}_{i-1}^Y; \boldsymbol{\theta}_Y), \tag{6}$$

where $\boldsymbol{y}_{i-1}$ is the word embedding of the previous word and $\boldsymbol{h}_i^Y$ is the hidden state of the GRU^Y at step i. $\boldsymbol{\theta}_Y$ represents all the parameters of the GRU^Y. Next, the hidden state $\boldsymbol{h}_i^Y$ is used to calculate the context vector $\boldsymbol{c}_i$ by an attention mechanism which will be detailed in the below. Finally, $\boldsymbol{h}_i^Y$ and $\boldsymbol{c}_i$ together is used to generate the word $\boldsymbol{y}_i$.

Gated Multi-Source Attention Mechanism

After the review encoder and product encoder, we get the review feature representation $\{\boldsymbol{h}_1^X, \boldsymbol{h}_2^X, \ldots, \boldsymbol{h}_n^X\}$ and the product feature representation $\{\boldsymbol{h}_1^T, \boldsymbol{h}_2^T, \ldots, \boldsymbol{h}_k^T\}$. In view of the fact that the word in response at step i is often only related to a small part of the review or product information, we utilize a gated multi-source attention mechanism to generate a high-level representation of the review part and product part. The proposed attention mechanism is introduced to make the decoder access useful review or product information in generation and it includes three parts: a review attention, a product attention, and a Gated Multimodal Unit (GMU). We also hope the generated response to cover all the issues in user's review, not just one. Therefore, we introduce the coverage vector $\boldsymbol{r}_i$ [27] in each time step, which maintains the sum of attention value over all previous steps and records what we have covered before step i. $\boldsymbol{r}_i$ is also used as an input to the review attention. Concretely, We first calculate the coverage vector $\boldsymbol{r}_i$ and then calculate the review context vector $\boldsymbol{c}_i^X$ by the review attention according to the following formulas:

$$\begin{aligned}
\boldsymbol{r}_i &= \sum_{j=1}^{i-1} \boldsymbol{a}_j^X, \\
e_{ij}^X &= (\boldsymbol{v}_X)^\top \tanh(\boldsymbol{W}_X \boldsymbol{h}_j^X + \boldsymbol{W}_{XY} \boldsymbol{h}_i^Y + \boldsymbol{W}_{Xr} \boldsymbol{r}_i), \\
\boldsymbol{a}_i^X &= \operatorname{softmax}(\boldsymbol{e}_i^X), \\
\boldsymbol{c}_i^X &= \sum_{j=1}^{n} a_{ij}^X \boldsymbol{h}_j^X,
\end{aligned} \tag{7}$$

where $\boldsymbol{v}_X, \boldsymbol{W}_X, \boldsymbol{W}_{XY}$ and $\boldsymbol{W}_{Xr}$ are learnable parameters and $\boldsymbol{a}_i^X$ is the attention weight at step i.

For the product attention, in addition to hidden state $\boldsymbol{h}_i^Y$, the review context vector $\boldsymbol{c}_i^X$ is also employed in the calculation of product attention, which allows the model to find relevant information in product according to the issue mentioned in review. Concretely, The calculation is made by the following equations:

$$\begin{aligned}
e_{ij}^T &= (\boldsymbol{v}_T)^\top \tanh(\boldsymbol{W}_T \boldsymbol{h}_j^T + \boldsymbol{W}_{TX} \boldsymbol{c}_i^X + \boldsymbol{W}_{TY} \boldsymbol{h}_i^Y), \\
\boldsymbol{a}_i^T &= \operatorname{softmax}(\boldsymbol{e}_i^T), \\
\boldsymbol{c}_i^T &= \sum_{j=1}^{k} a_{ij}^T \boldsymbol{h}_j^T,
\end{aligned} \tag{8}$$

where $\boldsymbol{v}_T, \boldsymbol{W}_T, \boldsymbol{W}_{TX}$ and $\boldsymbol{W}_{TY}$ are learnable parameters and $\boldsymbol{a}_i^T$ is the attention weight at step i.

To combine multiple sources of information, we use the Gated Multimodal Unit (GMU) [1] to learn fusion transformations and obtain the final context vector $\boldsymbol{c}_i$. The purpose of GMU is to find a feature representation based on a combination of data from different sources. The GMU learns to decide how different sources influence the activation of the unit using multiplicative gates. The equations governing the GMU are as follows:

$$\begin{aligned}
\boldsymbol{g}_i^X &= \tanh(\boldsymbol{U}_X \cdot \boldsymbol{c}_i^X), \\
\boldsymbol{g}_i^T &= \tanh(\boldsymbol{U}_T \cdot \boldsymbol{c}_i^T), \\
\boldsymbol{z}_i &= \sigma(\boldsymbol{U}_z \cdot (\boldsymbol{c}_i^X \oplus \boldsymbol{c}_i^T)), \\
\boldsymbol{c}_i &= \boldsymbol{z}_i * \boldsymbol{g}_i^X + (1 - \boldsymbol{z}_i) * \boldsymbol{g}_i^T,
\end{aligned} \tag{9}$$

where $\boldsymbol{U}_X, \boldsymbol{U}_T$ and $\boldsymbol{U}_z$ are learnable parameters and $\boldsymbol{z}_i$ is the gate that control the fusion transformations of $\boldsymbol{c}_i^X$ and $\boldsymbol{c}_i^T$.

Copy Mechanism

Through the above attention mechanism, we get the final context vector $\boldsymbol{c}_i$. In order to generate the next word, we need to define a score function. We use the generation score function defined by the following equation:

$$\boldsymbol{s}_i^g = \boldsymbol{W}_g(\boldsymbol{h}_i^Y \oplus \boldsymbol{c}_i) + \boldsymbol{b}_g. \tag{10}$$

where $\boldsymbol{W}_g$ and $\boldsymbol{b}_g$ are learnable parameters and $\boldsymbol{s}_i^g$ is the generation score.

To allow the decoder directly copies the words of review or product information, we also introduce two copying scores: the score of copying from review and the score of copying from product. That is,

$$\begin{aligned}
s_{ij}^{cr} &= \tanh((\boldsymbol{h}_j^X)^\top \boldsymbol{W}_{cr}) \cdot (\boldsymbol{h}_i^Y \oplus \boldsymbol{c}_i), \\
\boldsymbol{s}_i^{cr} &= \sum_{j=1}^{n} s_{ij}^{cr} \cdot \boldsymbol{1}_{\boldsymbol{x}_i},
\end{aligned} \tag{11}$$

and

$$\begin{aligned}
s_{ij}^{cp} &= \tanh((\boldsymbol{h}_j^T)^\top \boldsymbol{W}_{cp}) \cdot (\boldsymbol{h}_i^Y \oplus \boldsymbol{c}_i), \\
\boldsymbol{s}_i^{cp} &= \sum_{j=1}^{k} s_{ij}^{cp} \cdot \boldsymbol{1}_{\boldsymbol{t}_i},
\end{aligned} \tag{12}$$

where $\boldsymbol{W}_{cr}$ and $\boldsymbol{W}_{cp}$ are learnable parameters. $\boldsymbol{1}_{\boldsymbol{x}_i}$ is a one-hot vector of length V (V is the vocabulary size), among which the values are only with a single 1 in word $\boldsymbol{x}_i$ and all the others 0. The final score function is defined as

$$\boldsymbol{s}_i = \boldsymbol{s}_i^g + \boldsymbol{s}_i^{cr} + \boldsymbol{s}_i^{cp}, \tag{13}$$

and the probability of $\boldsymbol{y}_i$ is

$$P(\boldsymbol{y}_i | \boldsymbol{y}_{1:i-1}, \boldsymbol{X}, \boldsymbol{T}) = (\operatorname{softmax}(\boldsymbol{s}_i))_{\boldsymbol{y}_i}. \tag{14}$$

The maximum-likelihood loss at step t is given by:

$$\mathcal{L}_i^{ml} = -\log(P(\boldsymbol{y}_i | \boldsymbol{y}_{1:i-1}, \boldsymbol{X}, \boldsymbol{T})), \tag{15}$$

and the total loss is:

$$\mathcal{L}^{ml} = \sum_{i=1}^{m} \mathcal{L}_i^{ml}. \tag{16}$$

In training, we optimize the loss $\mathcal{L}^{ml}$ in the dataset.

3.2 Reinforcement Learning

During the training process, we feed the ground-truth output sequences to decoder and train the model by minimizing loss (16), while during inference decoder generates the next word given the previous predicted words. The errors will accumulate as decoder outputs the sequence, which is known as exposure bias problem [20]. Due to this problem, the maximum-likelihood training does not always produce the best results on evaluation metrics such as

	size	length of reviews	length of responses	number of records
Train	80000	39.96	72.67	15.22
Test	10000	39.76	72.64	15.27
Valid	10000	39.68	72.99	15.24

Table 2: Corpora statistics. We also give the average length of reviews and responses, and the average number of records of the product information.

Hyper-parameter	
vocab size	15000
word embedding	128
field embedding	32
position embedding	8
hidden size	256
batch size	16
learning rate	0.02
dropout rate	0.15
decay rate	0.8
gradient clipping	5

Table 3: Hyper-parameters used in all experiments (baseline models and our model).

ROUGE and BLEU and usually produces sub-optimal results with respect to these metrics.

In order to address this problem, we try to learn a policy that directly maximizes a specific evaluation metric, rather than minimizing the maximum-likelihood loss. Since the evaluation metric is non-differentiable, the traditional supervised learning methods cannot be used. Specifically, we employ self-critical sequence training (SCST) [21], a reinforcement learning algorithm, to achieve this optimization procedure. In SCST, we produce two output sequences: $\hat{Y}$, which is obtained by greedy decode, and Y^s, which is obtained by sampling from distribution at each decoding step. The reward function $r(Y)$ is defined as the specific metric, and we minimize the reinforcement loss:

$$\mathcal{L}^{rl} = (r(\hat{Y}) - r(Y^s)) \sum_{i=1}^{m} \log(P(\boldsymbol{y}_i^s | \boldsymbol{y}_{1:i-1}^s, \boldsymbol{X}, \boldsymbol{T})) \tag{17}$$

That is, if the sequence Y^s obtains a higher reward than baseline $\hat{Y}$, we will increase the probability of Y^s and decrease if it obtains a lower reward. Optimizing the specific metric encourages the model to focus more on global meanings, rather than on word-level discrepancy.

Therefore, the final loss is given by:

$$\mathcal{L} = \lambda_{rl} \cdot \mathcal{L}^{rl} + (1 - \lambda_{rl}) \cdot \mathcal{L}^{ml}, \tag{18}$$

where λ_{rl} ($0 \le \lambda_{rl} \le 1$) is a scaling factor that control the balance between $\mathcal{L}^{rl}$ and $\mathcal{L}^{ml}$.

4 EXPERIMENTS

4.1 The Taobao Dataset

We create the Taobao dataset from taobao.com, the largest online shopping website in China. There are 100K (review, product information, response) triples in the corpus which all belong to the category of Clothes. The product information is collected from the product page in the website and it is organized as a factual table. In the corpus, the average length of reviews is 39 words and the average length of response is 72 words. The average number of records in the table of product information is 15. The detailed corpora statistics is given in Table 2. We will release the dataset after the paper is published.

In our experiments, we use 80% of the data for training, 10% for validation and 10% for testing.

4.2 Evaluation Metrics

To evaluate our approach, we performed experiments on the Taobao dataset, and used the following evaluation metrics to assess the generation quality:

- **ROUGE**: ROUGE [14] is a widely used automatic evaluation metric in text summarization. The ROUGE score is obtained using the `pyrouge` package[2]. we report ROUGE- 1, ROUGE-2 and ROUGE-L in this paper.
- **METEOR**: METEOR[3] [6] has several features that are not found in other metrics, such as stemming and synonymy matching. It has proved to have good correlation with human judgement.
- **BLEU**: BLEU [18] is widely used in neural machine translation. It measures word overlap between the generated text and the ground-truth. The BLEU score is calculated using the `NLTK` package[4], in which the score is an average of BLEU-1~4.
- **Distinct**[5]: Distinct metric [13] measures how informative and diverse the generated responses are. Distinct-i represents the ratio of distinct i-gram in responses. We report Distinct-1 and Distinct-2 in this paper.
- **Human Evaluation**: We randomly sampled 100 cases and invited five volunteers to evaluate the generated responses of different models. For each triple (review, product information,response), volunteers are asked to give a score from {0,1,2}, which represents the quality of response. 0 means the response is irrelevant or disfluent with grammatical errors; 1 represents the response is related but not informative enough; and 2 indicates the response is not only related and natural, but also informative and interesting.

[2]https://pypi.org/project/pyrouge/
[3]http://www.cs.cmu.edu/~alavie/METEOR/
[4]https://www.nltk.org
[5]Since this metric is affected by the dataset size (as the size increases, the metric will tend to zero.), we only choose the first 200 cases to compute this metric.

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEU	Distinct-1	Distinct-2
Seq2Seq + Atte	29.93	7.05	21.98	11.95	4.22	5.73	14.49
Pointer-Generator	29.71	6.94	21.76	12.25	4.66	6.25	17.29
Copynet	30.57	7.17	22.37	12.39	4.46	4.89	12.24
Copynet + PI	33.91	11.99	26.35	14.44	10.75	8.66	25.15
Our Model	**35.06**	**13.66**	**27.72**	**15.06**	**12.30**	**8.70**	**25.90**

Table 4: Comparison with previous models using various automatic evaluation metrics. Note that the baseline "Copynet + PI" utilizes product information as well.

Model	2	1	0
Seq2Seq + Atte	68.2%	13.6%	18.2%
Pointer-Generator	68.8%	15.8%	15.4%
Copynet	69.2%	12.2%	18.6%
Copynet + PI	70.0%	16.4%	13.8%
Our Model	72.6%	15.4%	12.0%
Human Performance	82.4%	10.8%	6.8%

Table 5: Human evaluation of the five models. We show the percentage of each score. We also give the human performance in the last row.

4.3 Setups

In this work, we compare our model to the following four baseline models:

- **Seq2Seq + Atte**: "Seq2Seq + Atte" represents the standard Seq2Seq model with attention mechanism. The detailed description can be found in [4].
- **Pointer-Generator**: "Pointer-Generator" is proposed in [22], which augments "Seq2Seq + Atte" via pointing that can copy words from the source text and use coverage mechanism to eliminate repetition.
- **Copynet**: "Copynet" [9] integrates copying mechanism that is similar to the pointing in "Pointer-Generator", but the implementation is different.
- **Copynet + PI**[6]: "Copynet + PI" augments "Copynet" with the product information. The sequence of words $\{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_k\}$ which represents the table of product information is appended to the input review.

For the four baseline models and our model, we used mini-batch SGD with momentum to train these networks. The momentum was set to 0.9 and the initial learning rate was set at 0.02. After 10 epochs, the learning rate decayed every 4000 steps and the decay rate was 0.8. We employed the dropout technique with 15% dropout rate to avoid overfitting. To cope with the exploding gradient problem, gradient clipping was performed and we clipped the norm of the gradients at 5. The word embeddings were initialized by pre-trained embedding based on word2vec and then fine-tuned during training, but the field embedding and position embedding was randomly initialized. These three embedding size was set to 128, 32, and 8, respectively. For all models, we used two-layers GRU, and the hidden state size was set to 256. We used Jieba[7] as Chinese word segmenter and used a vocabulary of 15K words which was shared by the reviews and responses. These hyper-parameters are summarized in Table 3.

Since RL training is hard to converge from random initialization, we first train our model with the standard maximum likelihood loss (λ_{rl} is set to 0). After 30 epochs, we switch to RL training with λ_{rl} set to 0.99 and learning rate set to 0.0001. We choose the BLEU metric as the reward function. Our code will be made publicly available for reproducibility.

4.4 Automatic Evaluation

We first discuss the experimental results based on automatic evaluation metrics. Table 4 lists the results of four baseline models and our model. According to the results in Table 4, "Pointer-Generator" achieves better performance than "Seq2Seq + Atte" on all metrics except ROUGE and "Copynet" outperforms "Seq2Seq + Atte" on all metrics except Distinct 1~2, demonstrating the effectiveness of incorporating copy or pointing mechanism. Compared with other three baselines, "Copynet + PI" achieves significant improvement on all metrics. For example, it improves ROUGE-L by 3.98, BLEU by 6.29 and Distinct-2 by 12.91, compared with "Copynet". The results further verify our claim that product information is crucial and necessary for generating high-quality response, and the results on Distinct metric show that the diversity is increased greatly by incorporating product information. The last row of Table 4 shows the result of our model, which outperforms all four baseline models. For example, compared with "Copynet + PI", our model further improves ROUGE-L by 1.37, BLEU by 1.55 and Distinct-2 by 0.75, achieving start-of-the-art performance.

4.5 Human Evaluation

Table 5 gives the results of human evaluation. We give the percentage of each score, which is calculated by combining all the annotations together. We also calculated the variance of five volunteers' score. The result is 0.25, which demonstrates the fair inter-human agreements. From Table 5, it is clear that the models of incorporating product information ("Copynet + PI" and our model) generate much more relevant and informative responses (labeled as "2") and

[6]Copynet is the best-performed baseline (on most metrics) of the first three models, so we employ this method for PI integration.

[7]https://github.com/fxsjy/jieba

Percent	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEU	Distinct-1	Distinct-2
20%	30.68	8.06	22.88	12.70	5.88	7.43	21.32
40%	31.95	9.41	24.12	13.25	7.53	7.69	21.99
60%	32.92	10.62	25.30	13.84	8.91	8.08	22.51
80%	33.62	11.68	26.16	14.31	10.25	8.51	24.58
100%	**35.06**	**13.66**	**27.72**	**15.06**	**12.30**	**8.70**	**25.90**

Table 6: The results (based on various automatic evaluation metrics) of the proposed model when using various sizes (20%, 40%, 60%, 80%, 100%) of product information.

Model	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BLEU	Distinct-1	Distinct-2
Our Model	**35.06**	**13.66**	**27.72**	**15.06**	**12.30**	**8.70**	**25.90**
− RL	34.59	13.39	27.27	14.50	12.23	8.23	24.61
− Gate	34.76	13.33	27.45	14.83	11.90	8.19	24.60
− Copy	34.82	13.44	27.64	14.93	11.95	7.92	23.47
− Field	34.72	13.35	27.40	15.02	12.19	8.30	24.80

Table 7: Ablation studies on different components. The results (based on various automatic evaluation metrics) of removing (−) some components are given.

less universal responses (labeled as "1") than other baseline models. Among them, our model achieves the best performance. 72.6% of responses generated by our model are labeled as score 2, improving "Copynet + PI" by 2.6%, and only 12.0% of responses are labeled as score 0. We also evaluated the performance of responses written by human, and the results are shown in the last row of Table 5. Human achieves 82.4% on score 2 (which outperforms our model by 9.8%), 10.8% on score 1, and only 6.8% on score 0. This results indicate that there is still a huge room for improvement and we will continue to improve our model in the future.

4.6 Analysis on Product Information

In the proposed framework, we employed the product information in the review response generation, which is an essential contribution of this study. In this section, we explore the impact of the product information in the model. To investigate the impact of these data, we performed experiments on using various sizes (20%, 40%, 60%, 80%, 100%) of product information. For instance, 20% means that we randomly sample 20% of the whole of product information. The results of the proposed model (based on various automatic evaluation metrics) when using various sizes of product information are shown in Table 6. We can find that, with an increase in the amount of product information, all evaluation metrics gradually grows as well, which indicating the usefulness and importance of the product information in review response generation.

4.7 Ablation Study

To analyze the effect of each component, we conducted ablation studies on components of the proposed model. Table 7 lists the results of the ablation studies based on various automatic evaluation metrics. The performance of full model is listed in the first row of Table 7. We also report the results of removing (−) some components, such as the RL training (−RL in Table 7), the GMU gate mechanism (−Gate), the copy mechanism (−Copy) and the field representation (−Field). The first thing we can observe is that the RL training helps improve the performance on all metrics. For example, ROUGE-L drops to 27.27 from 27.72 and METEOR drops to 14.50 from 15.06 when we removed the RL training. The GMU gate mechanism is also crucial: all evaluation metrics drops if it is removed and replaced with a concatenation operation of $\boldsymbol{c}_i^X$ and $\boldsymbol{c}_i^T$. To test the importance of copy mechanism, we performed an experiment that removed the copy mechanism and we observed a drop, indicating that directly copying from review of product information can be very useful. From the results, we found that the copy mechanism is especially useful in improving the Distinct metrics. Distinct-1 drops to 7.92 from 8.70 and Distinct-2 drops to 23.47 from 25.90 when removing the copy mechanism. Moreover, the using of field representation is necessary for improving performance, which can effectively help the model in incorporating product information.

4.8 Case Study

To compare the generated responses between different models, we give two examples on Figure 5. The case in the left of Figure 5 is a positive review and the case in the right is a negative review. Below the reviews, We show the generated responses of two best baselines ("Copynet" and "Copynet+PI") and our model. Since we used a Chinese dataset, we give the original Chinese text and the English

评论：质量不错，款式好。很显身材，也很上档次。
Review: Good quality and good style. The clothes are figure flattering and also very high grade.

Copynet: 亲，感谢您对我们的支持，您的满意是我们最大的动力，我们会继续努力，为您提供更优质的服务，期待您的再次光临！

Copynet: Dear, thank you for your support, your satisfaction is our greatest motivation, we will continue to work hard to provide you with better service. Looking forward to your next visit!

Copynet + PI: 亲，感谢您对我们的支持与信任，您的满意是我们的最高追求。后期您有任何需要我们帮助的地方，随时欢迎垂询，我们将一如既往地为您效劳！能为您服务是我们莫大的荣幸！伊芙心悦旗舰店，祝亲生活愉快！

Copynet + PI: Dear, thank you for your support and trust, your satisfaction is our highest pursuit. Please feel free to contact us if you need our help. We will continue to help you! It is our great honor to be able to serve you! EVE NY wishes you a happy life!

Our Model: 感谢您的欣赏和喜欢，每一寸面料的裁剪，都伴随着汗水的淋漓。每一件衣服的出品，我们都会倾之付出百分百的努力和用心，希望能给您带来更舒适的穿着体验，希望伊芙心悦的时尚优雅能成为您一生的知己，给予您贴心的呵护。伊芙心悦旗舰店期待您的下次光临！

Our Model: Thank you for your appreciation and love. The cut of every inch of fabric is accompanied by sweat. Every piece of clothing, we will pay 100% effort and care. We hope to bring you a more comfortable wearing experience, and hope that EVE NY's stylish elegance can become your life's confidant and give you intimate pampering. EVE NY looks forward to your next visit!

评论：描述尺码和实际尺码差太多了，害得我买了大号！退换太麻烦，我给别人穿吧！
Review: There is too much difference between the descriptive size and the actual size, which makes me buy a large size! It's too much trouble to return. I will give it to others!

Copynet: 亲，我们的尺码都是标准码的哦，您可以参考下尺码表的哦，您的满意是我们最大的动力，我们会继续努力，为您提供更优质的产品，更优秀的服务。您的满意是我们最大的追求，祝您生活愉快！

Copynet: Dear, our size is the standard size. You can refer to the size chart. Your satisfaction is our greatest motivation. We will continue to work hard to provide you with better products and better service. Your satisfaction is our greatest pursuit, I wish you a happy life!

Copynet + PI: 感谢您选购我们的产品。由于每个人体型不一样，另外款式设计的不同，每个人穿着感觉会不一样的哦。有空常来看看哦~ 祝您购物愉快! [cachecache 旗舰店]

Copynet + PI: Thank you for purchasing our products. Because each body type is different, and the style design is different, everyone will feel differently. Visit my shop when you are free ~ I wish you a happy shopping! [cachecache flagship store]

Our Model: 亲爱的顾客，非常感谢您对 cachecache 旗舰店的关注。因每款的设计理念和面料弹性不同，相对应的尺码上是可能稍有差异的呢，建议您先测量自身基本尺寸，再参考页面尺寸信息进行购买，您也可事先咨询客服，祝您购物愉快！

Our Model: Dear customer, thank you very much for your attention to the cachecache flagship store. Because each style design and fabric elasticity are different, the corresponding size may be slightly different. It is recommended that you first measure your basic size and then refer to the size information for purchase. You can also consult customer service in advance. I wish you a happy shopping!

Figure 5: Case study, comparing the two best baselines with our model. We show the responses of one positive review in the left and the responses of one negative review in the right. Red denotes some product information ("EVE NY" and "cachecache" are *brand* name.) and blue denotes the trivial and universal responses.

translation. From the results, we can see that Copynet without product information tends to generate universal responses whether the review is positive or negative, like "*your satisfaction is our greatest motivation*" and "*we will continue to work hard to provide you with better service*". These universal responses will give users a very bad user experience, especially when users give a negative review. Instead, the models with product information (Copynet + PI and our model) generate more diverse and informative responses. For different reviews and different products, they can give different responses. For example, these responses not only give an explanation, but also mention the shop name (EVE NY and cachecache flagship store), which carry more information and give users more friendly feeling.

5 RELATED WORK

Seq2Seq Framework There has been quite a bit of research on text generation based on sequence to sequence framework. Sutskever et al. [26] first proposed the Seq2Seq framework, which maps sequences to sequences. Then, the attention mechanism [3] is proposed in neural machine translation to enhance the ability of capturing long term dependency, which is also widely used in other text generation tasks and made great success. The copy mechanism (or Copynet) [9] is proposed to incorporate copying into the Seq2Seq learning framework, which enable the model to copy words from the source text. A similar phenomenon is observable in human language communication. The Pointer-Generator [22] is close to the Copynet model, with some small differences. For example, they have an explicit switch probability to control the probability of generation or copy, while Copynet achieves the similar component through a shared softmax function. Different from Copynet and Pointer-Generator, our model achieves a copy mechanism that not only can copy text from review but also can copy from product information, which is very significant for the review response generation. Recently, some CNN-based [5, 8, 11, 28, 33] have been proposed to replace the recurrent structure in Seq2Seq and have achieved comparable performance while maintaining faster speed than RNN-based methods. For example, The Wavenet [28] and ByteNet [11] model used dilated convolutions to increase its receptive field in the encoder and decoder. In addition, the Transformer [29] model used multi-headed self-attention, which based solely on attention mechanisms, dispensing with recurrence and convolutions. We leave these as a future work.

Generation from Structured Data Generation text from structured data (e.g., a table) is also an important task in NLP. Lebret, Grangier, and Auli [12] proposed the representation of field using field name and position information. They generated text from Wikipedia infobox based a statistical n-gram model with local and global conditioning. Then, Liu et al. [15] proposed a structure-aware Seq2Seq method to model both content and structure of the table by local and global addressing, with a field-gating encoder and a dual attention mechanism. Sha et al. [25] presented an order-planning text generation model to capture the relationship between different fields and used such relationship to make the generated text more fluent and smooth. In our work, the product information is a structured data which is collected as a table. Different from them, the structured data is used as an auxiliary information and we need to model the interaction between the main text (review) and the structured text (product information). Wiseman et al. [32] found that even template based method exceed the performance of some neural models on some metrics in text generation conditioned on a small number of database records. We do not adopt template based method, since we find that it tend to generate universal responses. The work of Li et al. [7] also used product attributes in generation task. Different from us, they used attention-enhanced attribute-to-sequence model to generate product reviews, instead of review responses, for given attribute information. Our work also related to Xing et al. [34] and Wang et al. [30] which both incorporating auxiliary information in text generation. Xing et al. [34] proposed incorporating topic information into a Seq2Seq framework to generate informative and interesting responses for chatbots and Wang et al. [30] incorporated topic information into the convolutional Seq2Seq model in text summarization.

Generation with Reinforcement Learning Reinforcement learning trains an agent to maximize a reward function by interact with the external environment and has been used in many NLP tasks. Recently, many works have explored the use of reinforcement learning in text generation. When the metric that we want to optimize is not differentiable, the traditional supervised learning methods (such as optimizing the maximum-likelihood loss) cannot be used. Therefore, we can use an agent (in reinforcement learning algorithm) to perform discrete actions and obtain a reward. Ranzato et al. [20] first proposed a reinforcement learning method to train RNN-based sequence generation models, which based on the REINFORCE algorithm proposed by Williams [31]. Bahdanau et al. [2] used another reinforcement learning method, named actor-critic, to train neural networks to generate sequences. Rennie et al. [21] proposed self-critical sequence training to optimize the non-differentiable metric in image caption. Compared to previous supervised learning methods, their model leads to significant improvements. Then, Paulus, Xiong, and Socher [19] and Wang et al. [30] applied it to text summarization and demonstrate the superiority of these methods. In addition to being used in optimizing the non-differentiable metrics, reinforcement learning method can also be used in other ways. SeqGAN [35] addressed the differentiation problem in GAN by introducing the policy gradient methods when generating text, which bypasses the generator differentiation problem by directly performing gradient policy update. These works demonstrate the great potential of reinforcement learning method in NLP.

6 CONCLUSION

In this paper, we introduced the task of review response generation in e-commerce platforms, which is valuable for customer service providers, but has not yet been well-studied. To tackle this problem, we proposed a novel deep neural network model based on the Seq2Seq framework, which can incorporate external product information. A gated multi-source attention mechanism and a copy mechanism are utilized to leverage this product information. In addition, we used a reinforcement training technique to reduce the exposure bias problem. Experimental results based on both automatic evaluation metrics and human annotations demonstrated that the performance of the proposed model advances the state-of-the-art methods. A further analysis on each model component validates the effectiveness of our model and the case study demonstrates that our model can generate more informative and diverse responses than the baseline models.

A APPENDICES

A.1 More Detailed Statistics

We investigate the review data from taobao.com, the largest e-commerce website in China. Two product collections were randomly sampled, with 10K products for each. For the former one, the response rate of the negative reviews is above 80%, and the response rate for the latter one is below 20%. Statistics from these collections shows that the product selling volumes of these groups can be quite different, i.e., on average, 3545.79 items are sold per-product for high response rate group, and 3122.08 for low response rate group. In another experiment, we randomly sampled 500K products. For all the associated reviews, only 2.98% of the reviews and only 5.68% of the negative reviews have received the responses from sellers.

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by China National Key R&D Program (No.2018YFC0831105), National Natural Science Foundation of China (No.61751201, 61532011), and STCSM (No.16JC1420401, 17JC1420200).

REFERENCES

[1] John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A González. 2017. Gated Multimodal Units for Information Fusion. *arXiv preprint*

arXiv:1702.01992 (2017).
[2] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086* (2016).
[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[4] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.
[5] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083* (2016).
[6] Michael Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.
[7] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Vol. 1. 623–632.
[8] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *International Conference on Machine Learning*. 1243–1252.
[9] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1631–1640.
[10] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1700–1709.
[11] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099* (2016).
[12] Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural Text Generation from Structured Data with Application to the Biography Domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1203–1213.
[13] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 110–119.
[14] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out* (2004).
[15] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-Text Generation by Structure-Aware Seq2seq Learning. In *AAAI*. AAAI Press.
[16] Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421.
[17] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023* (2016).
[18] 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 on association for computational linguistics*. Association for Computational Linguistics, 311–318.
[19] Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304* (2017).
[20] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732* (2015).
[21] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *CVPR*.
[22] 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 (Volume 1: Long Papers)*, Vol. 1. 1073–1083.
[23] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models.. In *AAAI*, Vol. 16. 3776–3784.
[24] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues.. In *AAAI*. 3295–3301.
[25] Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2018. Order-Planning Neural Text Generation From Structured Data. In *AAAI*. AAAI Press.
[26] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.
[27] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling Coverage for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 1. 76–85.
[28] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio.. In *SSW*. 125.
[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.
[30] Li Wang, Junlin Yao, Yunzhe Tao, Li Zhong, Wei Liu, and Qiang Du. 2018. A Reinforced Topic-Aware Convolutional Sequence-to-Sequence Model for Abstractive Text Summarization. In *IJCAI*. ijcai.org, 4453–4460.
[31] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.
[32] Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052* (2017).
[33] Felix Wu, Ni Lao, John Blitzer, Guandao Yang, and Kilian Weinberger. 2017. Fast Reading Comprehension with ConvNets. *arXiv preprint arXiv:1711.04352* (2017).
[34] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic Aware Neural Response Generation.. In *AAAI*, Vol. 17. 3351–3357.
[35] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient.. In *AAAI*. 2852–2858.