Improving Text Generation for Product Description via Human Behaviour

Tong Guo

1Affiliation not available

February 20, 2024

Abstract

Text generation is an important method to generate high quality and available product description from product title. For the product description generation for online E-commerce application, the main problem is how to improve the quality of generated text. In other words, how we judge the quality of text. If all texts are already positive and available, then we find it impossible to manually judge which text is the better text for a product. So if we cannot judge which is a better text manually, we cannot improve the quality of generated text. In E-commerce, product description is to attract shoppers and improve sales. So we design a method to improve the quality of generated text based on user buying behaviour. Online result shows that our approach improve the sales of products by improving the text quality.
Abstract

Text generation is an important method to generate high quality and available product description from product title. For the product description generation for online E-commerce application, the main problem is how to improve the quality of generated text. In other words, how we judge the quality of text. If all texts are already positive and available, then we find it impossible to manually judge which text is the better text for a product. So if we cannot judge which is a better text manually, we cannot improve the quality of generated text. In E-commerce, product description is to attract shoppers and improve sales. So we design a method to improve the quality of generated text based on user buying behaviour. Online result shows that our approach improve the sales of products by improving the text quality.

1 Introduction

In recent years, the development of deep learning (DL) has brought breakthroughs on text generation. The sequence to sequence (Seq2Seq) models use encoder-decoder transformer (Vaswani et al., 2017) for better model flexibility. The most representative models of this type include T5 (Raffel et al., 2020) and BART (Lewis et al., 2020). In this paper, we adopt the T5 (Raffel et al., 2020) model to conduct our data-centric experiments.

In E-commerce, product description can attract shoppers and improve sales. But manually writing a successful product description is highly time-consuming. Text generation (Zhang et al., 2022; Prabhumoye et al., 2020) technologies play a crucial role in this range of applications.

Text generation has an input or a source sequence \( X \) and an output or a target sequence \( Y \) to be generated. In our product description generation tasks, \( X \) is the product title and \( Y \) is the product description. The examples are shown in Table 1.

The problem is to improve the quality of the generated text in E-commerce, then the following problem is how we define and judge what is high quality text. We find it impossible to manually judge which text is the better text for a product. To answer this problem, we define text that brings more sales to a product as better text. So we need to use user buying behaviour to judge which are better texts.

Then the problem is how to use user buying behaviour to select better quality text, the problem is that the text is displayed alongside the product, and we need to isolate the impact of the gain of the text on the product.

In summary, in order to improve the quality of generated description text for product. We face these problems:

1) We cannot judge which text is the better text for a product by human labeling. If we can’t judge and define text as good or bad, then we can’t optimize it.

2) We use user buying behaviour to judge better text. Then we need to isolate the gain impact of the text. Because text is displayed on the product. The product descriptions and products are bound together to be displayed to the users.

3) We need to design a complete solution to solve the above problems all together.

In this paper, we solve these problems and propose these contributions:

1) We use user buying behaviour to judge which text is better for a product, in order to solve the problem that we cannot judge it manually.

2) We train a sales prediction model upon the user buying logs of our E-commerce application. In order to isolate the gain impact of the text for the product, we use causal inference method.

3) We design a complete solution to continuously improve the quality of generated product description, guided by user behaviour.
<table>
<thead>
<tr>
<th>Input / Product Title</th>
<th>Output / Product Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birthday cake</td>
<td>Enjoy the taste and feel the beauty of the birthday.</td>
</tr>
<tr>
<td>Spicy chicken chunks</td>
<td>Crispy on the outside and tender on the inside.</td>
</tr>
<tr>
<td>Chicken soup with pig stomach</td>
<td>Nourishing and warming, with a long-lasting aftertaste.</td>
</tr>
<tr>
<td>Cheese hot dog</td>
<td>Full of cheese, can be pulled into strings.</td>
</tr>
<tr>
<td>Original mung bean cake</td>
<td>Made from peeled mung beans, pure and delicate.</td>
</tr>
<tr>
<td>Large cup of milk tea with two flavors</td>
<td>Long-lasting aftertaste, rich and fragrant.</td>
</tr>
</tbody>
</table>

Table 1: Examples of text generation for product description.

<table>
<thead>
<tr>
<th>Input / Product Title</th>
<th>Output-A</th>
<th>Output-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birthday cake</td>
<td>Enjoy the beauty of the birthday.</td>
<td>Make your birthday memorable.</td>
</tr>
<tr>
<td>Spicy chicken chunks</td>
<td>Crispy on the outside and tender on the inside.</td>
<td>It’s very addictive in one bite.</td>
</tr>
<tr>
<td>Cheese hot dog</td>
<td>Full of cheese, can be pulled into strings.</td>
<td>Cheese lovers eat it all at once.</td>
</tr>
</tbody>
</table>

Table 2: Examples of two outputs for the same input. We find it impossible to manually judge which text is the better text for the same input.

2 Method

The whole pipeline is shown in Figure 1. In this paper, we adopt the T5 (Raffel et al., 2020) model to conduct our text generation experiments. We adopt transformer (Vaswani et al., 2017) as our sales prediction model.

Our method contains 8 steps:

In Step-1, we get an initial dataset to train the T5 generative model. The initial dataset is constructed by query ChatGPT. We ask ChatGPT to write product description, input the product title as the prompts. We then remove the data in the training dataset that do not suitable to display online.

In Step-2, we use the T5 model to generate product descriptions for hundreds of millions the products.

In Step-3, we display the generated product descriptions on the products.

In Step-4, we collect the logs of sales and views of each product.

In Step-5, we train the sales prediction model based on the online logs. The detail is illustrated in the following sections.

In Step-6 and Step-7, we use causal inference to find out the best quality product descriptions in the logs. The detail is illustrated in the following sections.

In Step-8, we retrain the T5 model using the quality text identified of the last step. Then we do AB experiments to evaluate the performance of the generated product description of online App.

2.1 Initial Training Dataset Construction

This section corresponding to the Step-1 in Figure 1. We collect our initial training dataset by querying ChatGPT. Each prompt is formed by concatenating a product title. We ask ChatGPT to write descriptions for the products. We tried to add some product attributes as prompt, but most of the ChatGPT’s results do not relate to the product attributes. Table 1 shows the prompt examples and the ChatGPT’s results. Our T5 (Raffel et al., 2020) model trained on this initial dataset gets 88% available rate, under human evaluation.

2.2 Sales Prediction Model

After the generated product description has been displayed on the online products, The users view and buy the products. So now the problem is that we want to train a sales prediction model for products. The training target is:

$$RPM = \frac{N_{sales}}{N_{views}}$$

where $N_{sales}$ is the sales amount of product and $N_{views}$ is the viewed amount by users to this product.

The input features for the model include:
1) Product related features: product title, product tags, product history RPM.
2) Product description: Text tokens.

2.2.1 Training Target

We designed the training objective to capture the additional gain of text for product. So our training
objective is the absolute value of the RPM, and we use the regression loss.

\[ \text{Loss} = |RPM - \text{model\_output}| \]

2.3 Causal Inference

The role of causal inference in our approach is to be used to isolate the impact of text for product after having a trained sales prediction model. When we make a prediction, we input the product features and the text to get score \( A \), and only the product features to get score \( B \). The gain effect of the text for the product is \( A - B \). The detail framework of sales prediction and causal inference is shown in Figure 2.

3 Experiment

3.1 Manual Evaluation

The manual evaluation contains two parts: the generation available rate, the comparison of the two generation results. The available rate is to determine whether the generated text is available. We use human annotation to compute:

\[ \text{Rate} = \frac{N_{\text{good}}}{N_{\text{total}}} \]

where \( N_{\text{good}} \) is the available generated text number and \( N_{\text{total}} \) is the total texts that are human annotated.

The manual comparison of the Step-1 initial results and the optimised model results. The initial model results are corresponding to the Step-1 of the Figure 1. The optimised model results are the generated texts from the optimised model of the Step-8 of the Figure 1. We find it impossible to manually determine which of the two is more appropriate to be displayed on a product.

The available rate result is shown in Table 3. The comparison of the two generation result is shown in Table 4.

3.2 AB Experiments

We do AB experiments to evaluate the online performance of our method. We display the two product descriptions to two groups of users. Then we count the RPM of the two groups of users. The results show that the optimised texts improve the RPM by about 0.1%, compared to another group.

4 Discussion

In this section we illustrate the reason why we design our method. And we illustrate the baseline solution we compare.

4.1 Motivation

In order to improve the quality of the generated text, and to improve the RPM, we found that manual annotation can not achieve this, we look for supervisory signals from human behaviour of online App.

4.2 Baseline Solution

We first collect the results from ChatGPT to train our T5 model. We input product title and ask Chat-
Figure 2: The detail of the sales prediction model to isolate the impact of text for the same product. In the training data, i.e., the online logs, there are products that can display a product description and products that don’t display any product description. In the training phase, we use online logs as training dataset, which means that some products show text and some do not. In the prediction phase, we use the output score of the model with the input of text for the same product, and subtract the output score of the model without the input of text. We take the score obtained by subtracting as the score of text quality.

GPT to write product description. The problems with the ChatGPT results are that the available rate of text is 89% and 11% of the product description is not suitable for display. So we clean the dataset based on ChatGPT API and train the T5 model with more than 99% available rate of generated text. We use this generated results of our T5 model as the baseline, which is the Step-1 of Figure 1.

5 Relate Work

5.1 Text Generation

The pre-trained model based on Transformer (Vaswani et al., 2017) has greatly improved the performance in text generation. The learning objectives include masked language modeling (MLM) and causal language modeling (CLM). MLM-based Language Models include BERT (Devlin et al., 2018), ROBERTA (Liu et al., 2019). CLM-based Language Models include the GPT series works (Radford et al., 2018, 2019; Brown et al., 2020) and other decoder-only transformer models (Keskar et al., 2019). The sequence to sequence models (Sutskever et al., 2014) use encoder-decoder transformer (Vaswani et al., 2017) for better model flexibility. The seq2seq model is widely used in the field of text generation (Luong et al., 2014; Bahdanau et al., 2014). We adopt Seq2Seq models implement our text generation tasks. The most representative models of this type include T5 (Raffel et al., 2020) and BART (Lewis et al., 2020). In this paper, we adopt the T5 model to conduct our experiments. We compared T5 and GPT-2 (Radford et al., 2019) on the same dataset and ultimately chose T5.

5.2 CTR Prediction

Sales prediction task (Tsoumakas, 2019; Cheriyan et al., 2018) is to estimate future sales of products, which is same to the click through rate (CTR) prediction task (Chen et al., 2016; Guo et al., 2017). Sales prediction and CTR prediction both use users behaviour (click/view) as training target, which means that we collect the logs of online App to build the training dataset.

5.3 Causal Inference

The research questions that motivate most quantitative studies in the health, social and behavioral sciences are not statistical but causal in nature. Causal inference (Pearl, 2010, 2009) is to solve these problems. In our scenario, the product description is displayed on the product. That is, the product description, is the treatment that affects the sales of the corresponding product.

5.4 Text Quality Evaluation

The evaluation of text generation (Celikyilmaz et al., 2020; Zhang et al., 2019) is the task that evaluate of natural language generation, for example in machine translation and caption generation,
Table 3: The experiment result corresponding to the Figure 1’s pipeline steps. We adopt T5 (Raffel et al., 2020) to conduct our experiments.

<table>
<thead>
<tr>
<th>Model Version</th>
<th>Human Evaluation Available Rate</th>
<th>Training Dataset Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChatGPT API</td>
<td>89.0%</td>
<td>-</td>
</tr>
<tr>
<td>Step-1 Model, Before Data Cleaning</td>
<td>88.0%</td>
<td>300,000</td>
</tr>
<tr>
<td>Step-1 Model, After Data Cleaning</td>
<td>99.2%</td>
<td>285,000</td>
</tr>
<tr>
<td>Step-8 Model Results</td>
<td>99.3%</td>
<td>2,000,000</td>
</tr>
</tbody>
</table>

Table 4: Examples of two outputs for the same input. Output-B is the result before optimisation with our method. Output-A is the result after optimisation with our method.

<table>
<thead>
<tr>
<th>Input / Product Title</th>
<th>Output-A</th>
<th>Output-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birthday cake</td>
<td>A taste of home.</td>
<td>Make your birthday memorable.</td>
</tr>
<tr>
<td>Spicy chicken chunks</td>
<td>Super, super tasty.</td>
<td>It’s very addictive in one bite.</td>
</tr>
<tr>
<td>Cheese hot dog</td>
<td>A big bite of meat and a lot of fun.</td>
<td>Cheese lovers eat it all at once.</td>
</tr>
</tbody>
</table>

requires comparing candidate sentences to annotated references. In our scenario, however, we are unable to manually evaluate the impact of the quality of the generated product description on product sales. So we do AB experiments to count whether the generated text leads to an increase in product sales or not, to judge whether the quality of text is improved.

6 Conclusion

How to improve the quality of generated text is a very critical issue, as manual annotation cannot judge the quality of generated text. If manual annotation cannot judge the quality of the generated text, then we cannot optimise the text generation to a better quality direction. On the other hand, if we can find a method to judge the quality of the generated text, then we can continuously optimise it. This paper we find supervised signals in the E-commerce scenario that can continuously optimise the quality of generated text. We have developed a complete solution and sales of our App have been boosted.

References


