Abstract

Product description generation is a challenging and under-explored task. Most such work takes a set of product attributes as inputs then generates a description from scratch in a single pass. However, this widespread paradigm might be limited when facing the dynamic wishes of users on constraining the description, such as deleting or adding the content of a user-specified attribute based on the previous version. To address this challenge, we explore a new draft-command-edit manner in description generation, leading to the proposed new task—controllable text editing in E-commerce. More specifically, we allow systems to receive a command (deleting or adding) from the user and then generate a description by flexibly modifying the content based on the previous version. It is easier and more practical to meet the new needs by modifying previous versions than generating from scratch. Furthermore, we design a data augmentation method to remedy the low resource challenge in this task, which contains a model-based and a rule-based strategy to imitate the edit by humans. To accompany this new task, we present a human-written draft-command-edit dataset called E-cEdits and a new metric “Attribute Edit”. Our experimental results show that using the new data augmentation method outperforms baselines to a greater extent in both automatic and human evaluations.¹

1 Introduction

In E-commerce, controllable text generation plays an essential role in generating attractive and suitable product descriptions (Shao et al., 2021). These automatic description generation methods bring significant increases in writing efficiency and cost savings when facing billions of product data (Zhang et al., 2019).

Most recent works depend on the single-pass paradigm to design the description generation manner (Li et al., 2020a; Chan et al., 2019; Shao et al., 2021). In details, the generation model takes a set of inputs, which include the key selling points of a product (i.e., a set of attributes (Shao et al., 2021)) and various forms of grounding, such as titles (Zhang et al., 2019), customer reviews (Zhan et al., 2021), or knowledge base (Chen et al., 2019). Then, it outputs the final description, without considering user feedback (i.e., in a single pass).

Despite the success of the above studies, there is still a major limitation in this single-pass paradigm—it fails to interact with users and flexibly refine the generated description. Specifically, users’ expectations about the content might change after receiving the first version of description. For example, they may want to add or delete the content of a specific product attribute to obtain a more appropriate description. Unfortunately, it is inflexible for the single-pass paradigm to address this situation. On the one hand, existing models have to regenerate a description from scratch, even if

¹Our code and dataset have been uploaded as supplementary materials, which will be released upon the acceptance.
needing a very few words changes from the previous version. On the other hand, utilizing manual post processing is time-consuming and often prohibitively expensive (Green et al., 2013), since the edit operation includes finding the right place from the entire description, then rewriting the attribute-relevant content while fine-tuning context to keep readability.

To reach the ideal goal of generating descriptions interactively, we propose a new task to approximate the condition—controllable E-commerce description editing. In short, we allow users to flexibly modify the previous description (hereinafter known as draft) in a draft-command-edit manner. In particular, users input a specific command with an attribute (e.g., deleting or adding the attribute-relevant content in draft), the model then generates a new description (i.e., edit) based on the command, draft, and grounding (e.g., product title, product property). This paradigm would be easier and more practical to meet the new needs of users (Faltings et al., 2020), while making the first attempt to high-volume processing on user-oriented description editing.

The key challenge of this task is that the rich supply of alignment data between draft and edit would be naturally inaccessible, which may cause big difficulties for these data-driven generation models. To overcome this low resource limitation, we propose a data augmentation method to automatically generate draft-edit data pairs. In more detail, we design two strategies to imitate the edit by humans. One of them uses a filling-in-the-blank generation model to approximate the human edit that removes the attribute-relevant tokens and slightly modifies the context to keep readability. Another one is based on the rules to imitate the edit that directly deletes the attribute tokens in the content.

Besides, we introduce a new draft-command-edit dataset called E-cEdits and a novel metric “Attribute Edit” to evaluate our method. E-cEdits is created by humans via crowdsourcing in Anonymity E-commerce scenario. As Figure 1 shown, we first crawl data from the platform, then the human annotators are asked to edit the description until it excludes the content about a pre-specified attribute. In the end, there are 9,000 <attribute, command, grounding, draft, edit> 5-tuples from 733 product categories. In addition, “Attribute Edit” aims to examine whether an attribute has been edited. In details, it computes a fuzzy matching score between the input attribute and the model output, evaluating whether the content of the user-specified attribute appears (for adding commands) / disappears (for deleting commands) in final results. It is simple but effective, and our experiments demonstrate that “Attribute Edit” metric is significantly correlated with human evaluation.

The main contributions of this work could be summarized as follows:

- We propose a challenging new text generation task in E-commerce, which allows the model to flexibly modify the description constrained by users’ dynamical requirements. This paradigm could provide a novel insight to design a user-oriented generation manner in controllable text generation.
- Responding to the key challenge of low resource in this task, we propose a new automatic data augmentation method to approximate human edit, which automatically generates pseudo data. Experiments on the E-cEdits dataset show that our system significantly outperforms baselines in automatic and human evaluations.
- To accompany this task, we release an E-commerce text editing dataset E-cEdits and design a novel metric “Attribute Edit”.

2 Related Work

In this section, we first analyze the related works on text generation in E-commerce. Then, we review some representative works which introduce text editing into some text generation scenarios.

2.1 Text Generation in E-commerce

Recently, various attempts have been made in E-commerce text generation, such as title generation (Mane et al., 2020), summarization (Li et al., 2020b), dialogue (Zhang et al., 2020a), and answer generation (Gao et al., 2021). The most related task to us is product description generation. Apex (Shao et al., 2021) based on a Conditional Variational Autoencoder generates a description from a set of attributes. FPDG (Chan et al., 2019) considers the entity label of each word and increases the fidelity of the descriptions. KOBE (Chen et al., 2019) takes a variety of factors into account while generating
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description $\hat{x}$</th>
<th>Description $x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>shape: column</td>
<td>brass magnetic clasps, column, silver size: about 8mm wide, ..., just add to the end of your diy bracelets crimp in the hole.</td>
<td>brass magnetic clasps, silver size: about 8mm wide, ..., just add to the end of your diy bracelets crimp in the hole.</td>
</tr>
<tr>
<td>flavor: green-apple flavor</td>
<td>you will receive green-apple flavor, a must have for braces, ..., package including: 10 boxes ortho wax.</td>
<td>you will receive a must have for braces, ..., package including: 10 boxes ortho wax.</td>
</tr>
<tr>
<td>club type: hybrids</td>
<td>brand new aftermarket adapter for taylormade hybrids, ..., they have 1.5 degrees of loft adjustment.</td>
<td>brand new aftermarket adapter, ..., they have 1.5 degrees of loft adjustment.</td>
</tr>
<tr>
<td>nintendo model: nintendo switch</td>
<td>charging data cable for nintendo switch type c usb charger, ..., 1pcs x charging cable.</td>
<td>charging data cable for the one that uses type c usb charger, ..., 1pcs x charging cable.</td>
</tr>
</tbody>
</table>

Table 1: Example description edits of E-cEdits. The unmodified contents are omitted and the modified properties are colored. Best viewed in color.

descriptions, including product aspects, user categories, and knowledge base. The previous methods have their applicantive advantages and insurmountable disadvantages: for example, they consider the generation of description under the one-pass setting from scratch. However, the users’ needs for constraining the description could be dynamic. In contrast, we provide a new generation paradigm regarding the process in a draft-command-edit manner, which significantly drops the difficulty of the description generating by taking advantage of previous versions.

2.2 Text Editing

Dating back to the period of rule-based postediting (Knight and Chander, 1994), text editing has long been investigated for text generation. According to the differences in the ultimate goal, it can be roughly divided into two types: (1) Refining the sentences to be more fluency and factually grounded; (2) Adding or modifying the content. The first type has a set of different settings, such as post-editing (Herbig et al., 2020; Mallinson et al., 2020), grammatical error correction (Zhao and Wang, 2020; Wan et al., 2020), and paraphrasing (Goyal and Durrett, 2020; Siddique et al., 2020). Our new task is more relevant to the second type, in which editing text based on prototypes has achieved promising performance. For example, Guu et al. (2018) sample prototype from the training corpus. FACTEDITOR (Iso et al., 2020) creates a fact-based draft by rules as the model input in two data-to-text tasks. Faltings et al. (2020) crawl data from Wikipedia’s revision histories to form a Draft-Edit pair, and generate text according to the command in a progressively adding manner. However, directly incorporating the methods above into E-commerce is less portable. It is because the text editing manner of these methods is based on continuing writing (i.e., generating new sentences after the current text), which might be hard to modify previous content according to users’ wishes. In comparison, our task provides more flexible operations—adding and deleting, which takes modifying and adding content of the description both into account. Meanwhile, the editing object product attribute is the central theme of a product description, which promotes our task more adaptable to the application requirements in E-commerce (Petrovski and Bizer, 2017).

3 Preliminaries

In this section, we introduce our new task and elaborate the benchmark E-cEdits. To ease of presentation, we start from formalize the new task in § 3.1. Then we give a detailed creation of the presented dataset E-cEdits in § 3.2.

3.1 Task Definition

The controllable E-commerce description editing task is defined as follows: given an attribute $a$, a command $\mu$ and the specified forms of grounding $g$, the system generates a new description $x$ based on the previous version $\hat{x}$. In our specific settings to instantiate this task, the command $\mu$ is defined as adding or deleting the content of attribute $a$ from the description $\hat{x}$ while keeping readability. Meanwhile, the type of attribute-relevant content is various. It could be a word, a phrase, or a clause when appearing in the description. In addition, grounding is the supplementary information about the product, such as titles. In sum, the editing process is $\hat{x} \rightarrow x$, given $a$, $\mu$, and $g$.  

---

### 3.2 E-cEdits

It is tough to obtain \( \hat{x} \) for \( x \) from the E-commerce platform, since the history version for attributes editing is difficult to collect. To accompany this task and evaluate the effectiveness of the proposed method, we present a high-quality dataset containing 9,000 {\textit{draft-command-edit}} tuples in the English E-commerce domain, which is wholly written by humans via crowdsourcing in the Anonymity E-commerce platform. To be concrete, we ask each annotator to remove the content of a pre-specified attribute from the complete description, which is consistent with the deleting editing in the task definition.\(^3\) When considering the adding editing, data tuples can be easily obtained by exchanging the deleted editing samples’ source and target descriptions. It is worth mentioning that the editing operation of humans follows the “minimum modification principle”. This principle means that the annotator should remove the relevant content, may add punctuation or a few words to keep the readability. In addition, the third-party inspectors examine 200 random samples from the editing data for data quality assurance and make sure the pass rate is more than 95%. Finally, each sample of E-cEdits contains 5-tuple <attribute, command, grounding, draft, edit>. In the implementation, the command includes two signals, “[ADD]” and “[DEL]”, which denote adding and deleting commands, respectively. Meanwhile, we choose product title and category as the grounding following Zhang et al. (2019).

#### Statistical Analysis

Table 2 gives a statistical overview of our dataset. More concretely, the mean length difference between \( \text{draft} \) (i.e., \( \hat{x} \)) and \( \text{edit} \) (i.e., \( x \)) suggests that the words changing in editing is slightly. Meanwhile, descriptions come from 733 categories, and various types mean our dataset could approximate the real condition. Finally, the mean length of the attribute implies that it usually includes only a single word or a short phrase. As a result, the minimal information offered by attributes may bring algorithms difficulties to generate a description from scratch.

**Examples** The typical edit examples of E-cEdits are illustrated in Table 1. The phenomena can be mainly divided into two categories: 1) Deleting the attribute-relevant content. For instance, the annotator may delete the keywords (row 2), the phrase (row 3), or the clause (row 1); 2) Replacing the attribute words with attribute-free ones (row 4) to keep the text flowing. We can see that the editing contains various forms and appears in different positions, which will present the editing system in the low-resource settings with a great challenge.

### 4 Method

In this section, we elaborate our data augmentation method and the description editing model. To ease of presentation, we start from toy examples to illustrate the overview of the data augmentation method in § 4.1, and give a detailed explanation of the implementation. Then, we present the editing model in § 4.2.

#### 4.1 Automatic Data Augmentation

To remedy the low resource challenge in this task, we design a data augmentation method to imitate the edit by humans. Although {\textit{draft-command-edit}} data pairs are difficult to obtain naturally, a large number of descriptions and corresponding attributes can be easily collected from E-commerce platforms. Thus, we consider strategically removing the content of a pre-specified attribute from the description (i.e., the deleting editing), which is similar to the dataset building method of humans editing. After that, we obtain the adding editing sample by exchanging the source and target descriptions in deleting, as mentioned in § 3.2.

**Model-based Strategy** Our basic idea is to mask the content of a pre-specified attribute in description with a signal “[FILL]” then use a filling-in-the-blank model to generate attribute-free content on that position. Therefore, we get a \textit{draft} for each description while keeping readability. The workflow is shown in the blue flow of Figure 2. For each description, we extract a word or a phrase, then replace it with a mask token “[FILL]”. After that, we fine-tune a pre-trained Seq2Seq model (e.g., ProphetNet (Qi et al., 2020)) to reconstruct...
the description (i.e., the “Pretrained model” part in Figure 2). In the inference, we use a phrase matching tool based on the Levenshtein Distance to mask the content of a pre-specified attribute with “[FILL]”. Then, we constrain the decoding space by removing the attribute tokens in vocabulary while generating. As a result, the model can generate an attribute-free description.

It is worth mentioning that we develop multiple policies for deciding which tokens or phrases should be masked with “[FILL]”. First, we use the TF-IDF score at the token level to choose and mask an essential word in each description (except for stop words and punctuation). The TF-IDF score could provide the uniqueness and local importance of a word at the corpus level (Zhang et al., 2020b).

Second, we design three masking approaches to increase the filling types’ variety at the phrase level, aiming to approximate human editing situations. Concretely, it includes random, conjunction-based, and attribute-based masking (the blue block “Phrase Level”). In randomly masking, we randomly choose 2-5 word pieces to mask each description, referring to the length statistic results. Meanwhile, we consider masking attribute-relevant words with phrase fuzzy matching tool thus models can enjoy benefits to better deal with attributes. In addition, we randomly mask the clauses connected with coordinating conjunctions as it is a typical form when multiple attributes appear in one sentence. Finally, we collect 1.2 million data pairs in total for model training. The proportion of each type in the training data set is: 50% for token level and 50% for phrase-level (50% attribute-based masking, 30% conjunction-based masking, and 20% randomly masking).

**Rule-based Strategy** We design two ways to imitate the editing type of directly deleting attribute-relevant content in a description. On the one hand, we directly remove the sentence in a description if it only contains one attribute (the “Sentence Level” block in the figure). On the other hand, as removing adjectives does not affect the sentence integrity, we also fall the attributes of adjectives into this category (the “For adjective” block).

**4.2 Model**

For the description edit task, we use the standard auto-regressive sequence to sequence models as test beds, thus various generation models can be
where \( \theta \) is the model parameters.

In implementation, we use the pre-trained Seq2Seq model ProphetNet (Qi et al., 2020) as the backbone, which is effective on several text generation tasks (Liu et al., 2020). To adapt the ProphetNet in our task, we follow Gururangan et al. (2020) to continue training it in three steps. Firstly, to adapt ProphetNet to the E-commerce domain, we collect 24 million grounding E-commerce titles to continue pre-train ProphetNet in a denoising sequence-to-sequence task. After that, we obtain a pre-trained E-commerce domain Seq2Seq model called ProphetNet-E. Secondly, we train ProphetNet-E in text editing task using our automatic augmentation dataset as described in § 4.1, which contains 600 thousand samples. Finally, we train the model on the E-cEdits dataset. As shown in Figure 3, in the editing task, it has to be mentioned that we concatenate all of the model’s input (\( a, \mu, g, \) and \( \hat{x} \)) by the separator signal “[SEP]” to adapt the models for our task.

5 Experiments

5.1 Experimental Settings

For the E-cEdits dataset, we randomly sample 4,000/1,000 pairs for training/validation, and the remaining 4000 for testing. All of the models are implemented based on Fairseq (Ott et al., 2019), and the specific parameters setting for each model can be found in § 5.2. We set the max training epoch to 10 for each model. In the inference step, the beam size and length penalty are set to 4 and 1.2 respectively, to calculate the main results without post-processing.

5.2 Baselines

Transformer is the most commonly used sequence to sequence model. We follow the hyperparameters of standard Transformer (Vaswani et al., 2017).

MASS (Song et al., 2019) uses the encoder-decoder framework to reconstruct a sentence fragment given the remaining part of the sentence. We use the largest released pretraining model “MASS-middle-uncased” trained on Wikepika and BookCorpus. It contains six layers (embedding/hidden size 1024 and 16 head for each attention layer) for both encoder and decoder.\(^5\)

ProphetNet (Qi et al., 2020) introduces an n-stream self-attention mechanism and a self-supervised objective named future n-gram prediction. Two versions of ProphetNet are used, and the main difference is the source of the training dataset. “ProphetNet” in Table 3 is trained on English Wikepika and BookCorpus (16GB in total), while “ProphetNet-E” is continue trained on E-commerce text. Both of two models include a 12-layer encoder (decoder) with 1024 embedding/hidden size and 4096 feed-forward filter size.\(^6\)

5.3 Evaluation Metrics

Automatic Evaluation Following the automatic evaluation methods in both text editing (Faltings et al., 2020; Iso et al., 2020) and description generation in E-commerce (Chen et al., 2019; Chan et al., 2019), we first use BLEU (Papineni et al., 2001) and ROUGE (Lin, 2004) to keep in line with previous works. The BLEU score is calculated by the built-in function in Fairseq, while the ROUGE score is calculated by the “files2rouge” tool.\(^7\)

While ROUGE and BLUE could evaluate fluency of the generated description, there is no way to explicitly examine whether an attribute has been edited, which is one of the key points in this task. To tackle this problem, we propose a new evaluation indicator “Attribute Edit”, which computes the fuzzy matching score between the input attribute and the model output (the matching tool is introduced in § 4.1, and score range 0-100). It is worth mentioning that this indicator is significantly correlated with human evaluation (details

\(^5\)https://github.com/microsoft/MASS
\(^6\)https://github.com/microsoft/ProphetNet
\(^7\)https://github.com/pltrdy/files2rouge
Table 3: Performances of our system and baselines on E-cEdits dataset in terms of Fluency (BLEU and ROUGE) and Attribute relevance (Attribute Edit). R-1, R-2, and R-L denote ROUGE-1, ROUGE-2, and ROUGE-L, respectively. B-4 represents four gram BLEU score. All of ablations are based on ProphetNet-E.

<table>
<thead>
<tr>
<th>Item</th>
<th>Model</th>
<th>BLEU</th>
<th>ROUGE</th>
<th>Attribute Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ADD</td>
<td>DEL</td>
<td>ALL</td>
</tr>
<tr>
<td>Baselines (row 1-3)</td>
<td>Transformer</td>
<td>56.32</td>
<td>55.57</td>
<td>0.37</td>
</tr>
<tr>
<td></td>
<td>MASS</td>
<td>58.47</td>
<td>90.00</td>
<td>-15.76</td>
</tr>
<tr>
<td></td>
<td>ProphetNet</td>
<td>59.33</td>
<td>87.60</td>
<td>-14.43</td>
</tr>
<tr>
<td></td>
<td>only Grounding</td>
<td>61.72</td>
<td>85.77</td>
<td>-12.03</td>
</tr>
<tr>
<td></td>
<td>only Command</td>
<td>61.08</td>
<td>87.24</td>
<td>-13.08</td>
</tr>
<tr>
<td></td>
<td>no Data Augmentation</td>
<td>61.24</td>
<td>85.97</td>
<td>-12.36</td>
</tr>
<tr>
<td></td>
<td>Our system</td>
<td>87.29</td>
<td>58.09</td>
<td>14.60</td>
</tr>
</tbody>
</table>

Table 4: Human Evaluation for Mass and our system. Note that the higher of “DEL” score represents the better quality as opposed to the automatic evaluation. The significance test is carried out for each group by the Two-Sample t-Test, all of the p values are less than 0.05.

<table>
<thead>
<tr>
<th>Model</th>
<th>Fluency ↑</th>
<th>Attribute-relevant ↑</th>
<th>Overall ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADD</td>
<td>DEL</td>
<td>ALL</td>
</tr>
<tr>
<td>MASS</td>
<td>3.61</td>
<td>3.75</td>
<td>3.68</td>
</tr>
<tr>
<td>Our system</td>
<td>3.93</td>
<td>3.95</td>
<td>3.94</td>
</tr>
</tbody>
</table>

Table 5: The coefficients of Pearson correlation with human evaluation. “LD” denotes Levenshtein distance.

<table>
<thead>
<tr>
<th>Item</th>
<th>Attribute Edit</th>
<th>Character-level LD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>0.92 (p-value&lt;0.0001)</td>
<td>-0.01 (p-value=0.96)</td>
</tr>
<tr>
<td>DEL</td>
<td>0.89 (p-value&lt;0.0001)</td>
<td>0.17 (p-value=0.23)</td>
</tr>
</tbody>
</table>

5.4 Main Results

We evaluate the performance of our system and baselines on E-cEdits dataset and further provide ablations. We report the main result on Table 3 and the human evaluation can be found in Table 4, from which we can make the following conclusions:

1. **Our system consistently outperforms baselines both in automatic evaluation and human evaluation.** For automatic evaluation (as shown in Table 3), our system outperforms all baselines both in fluency (BLEU and ROUGE) and attribute relevance (Attribute Edit). For example, comparing with MASS that gets the highest 89.30 BLEU and 94.72/92.05/94.69 ROUGE scores in text editing baselines (row 2), our system still beats MASS with 91.78 BLEU and 96.52/94.01/96.28 ROUGE score (row 7). Especially in Attribute Edit, using the proposed data augmentation method can significantly improve the editing model compared with other baselines. We choose MASS to compare with our system in human evaluations because it is the best baseline in both ROUGE and BLEU. As shown in Table 4, our system also outperforms MASS in both fluency and attribute relevance. Consistent with the automatic evaluation, these two models show significant differences in “Attribute-relevant” scores (2.05 vs. 3.95).

2. **Both command and grounding play important roles in the controllable text editing task.**

...
We also explore the impact of grounding and command on model performance. As shown in ablations (row 4 and row 5), using explicit command may improve performance in fluency, while using the grounding benefits model in better attribute editing. Meanwhile, ProphetNet-E outperforms all baselines (row 6) when in the “no Data Augmentation” condition. That is, using E-commerce text for pretraining could better adapt ProphetNet to this task (compared with row 3).

5.5 Further Analysis

Attribute Relevance Evaluation We further compute Pearson correlation coefficient between the human score and our “Attribute Edit” score, to verify it can effectively evaluate whether models carry out an editing operation. Meanwhile, we also choose the character-level Levenshtein distance as the baseline, which is widely used in judging the two sentences’ similarity (Snover et al., 2006). Table 5 illustrates that there is a significant statistical correlation between the proposed “Attribute Edit” score and human evaluation. In addition, the character-level Levenshtein distance is irrelevant with human evaluation as all of the p-values greater than 0.05.

Case Analyze Table 6 illustrates four examples of description editing results with our edit system. Especially, we can see that the operation of adding and deleting is not just simply copying or removing all the words in the pre-specified attribute, as finding an appropriate position in draft for editing operation is one of the challenges in this task. For example, with deleting command, sample 4 needs to remove the attribute “product type: stopper”. Our model not only removes the word “stopper”, but also the relevant content “24mm width”, which keeps the readability of the Edit version.

Table 6: Examples for adding and deleting in our final results, where the colored words are matched with attributes.

| Attribute: Chain style: handmade chains | Command: Adding | Draft: Beads for necklaces bracelets making, with silver iron eye pins, mixed stone... | Model output: Handmade stone chips beads chains for necklaces bracelets making, with silver iron eye pins, mixed stone... |
| Attribute: Item name: original rear camera module replacement part | Command: Adding | Draft: For sony xperia xa ultra compatible: for sony xperia xa ultra (only fit for xa ultra, not fit for xa1 ultra) features: original high... | Model output: Original rear camera module replacement part for sony xperia xa ultra (only fit for xperia ultra), not fit for xa1 ultra, features: original high... |
| Attribute: Suitable: abs systems | Command: Deleting | Draft: Feature: automatic brake... and magnetic reservoir base to prevent accidental spillage. Suitable for use with abs systems. Supplied with range of... | Model output: Feature: automatic brake... and magnetic reservoir base to prevent accidental spillage. Supplied with range... |
| Attribute: Product type: stopper | Command: Deleting | Draft: We have our own button factory... bulk order have more discount. Item description: plastic stoppers: 24mm width * 23mm height. | Model output: We have our own button factory... bulk order have more discount. |

6 Conclusion

In this paper, we propose a new controllable text editing task allowing users flexibility to constrain the attribute-relevant content of the product description by commands in a draft-command-edit manner, and introduce a high-quality draft-command-edit dataset E-cEdits written by humans. Meanwhile, in response to the low resource condition—the key challenge in this task, we design a data augmentation method that contains two strategies to generate pseudo data pairs. Experiments demonstrate that our method significantly and consistently outperforms baselines both in automatic evaluation and human evaluation. In sum, as a new attempt, we tentatively give a simple but effective implementation of product description editing, successfully approximate the ideal goal of generating descriptions interactively. Thus in the future, such a paradigm deserves a closer and more detailed exploration. Therefore, we will investigate to design this interactive generation manner in a more superior way.
References


