Automatic Controllable Product Copywriting for E-Commerce

Xiaojie Guo
xguo7@gmu.edu
JD.COM Silicon Valley Research
Center
USA

Qingkai Zeng*
qzeng@nd.edu
University of Notre Dame
USA

Meng Jiang
mjiang2@nd.edu
University of Notre Dame
USA

Yun Xiao
xiaoyun1@jd.com
JD.COM Silicon Valley Research
Center
USA

Bo Long
bo.long@jd.com
JD.COM
China

Lingfei Wu
lwu@email.wm.edu
JD.COM Silicon Valley Research
Center
USA

ABSTRACT

Automatic product description generation for e-commerce has witnessed significant advancement in the past decade. Product copywriting aims to attract users' interest and improve user experience by highlighting product characteristics with textual descriptions. As the services provided by e-commerce platforms become diverse, it is necessary to adapt the patterns of automatically-generated descriptions dynamically. In this paper, we report our experience in deploying an E-commerce Prefix-based Controllable Copywriting Generation (EPCCG) system into the JD.com e-commerce product recommendation platform. The development of the system contains two main components: 1) copywriting aspect extraction; 2) weakly supervised aspect labelling; 3) text generation with a prefix-based language model; and 4) copywriting quality control. We conduct experiments to validate the effectiveness of the proposed EPCCG. In addition, we introduce the deployed architecture which cooperates the EPCCG into the real-time JD.com e-commerce recommendation platform and the significant payoff since deployment. The codes for implementation are provided at https://github.com/xguo7/Automatic-Controllable-Product-Copywriting-for-E-Commerce.git.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Natural language generation**; *Ranking*; *Neural networks*; • **Information systems** → *Online shopping*; • **Applied computing** → *E-commerce infrastructure*.

KEYWORDS

Product description generation; text generation; e-commerce

---

1 INTRODUCTION

Informative product copywriting is critical for providing a desirable user experience on an e-commerce platform. Different from brick-and-mortar stores where salespersons can hold face-to-face conversations with customers, e-commerce stores heavily rely on textual and pictorial product descriptions which provide product information and eventually promote purchases [7]. Accurate and attractive product descriptions help customers make informed decisions and help sellers promote products. Traditionally, human copywriters perform product copywriting, which exposes significant limitations to match the growth rate of new products.
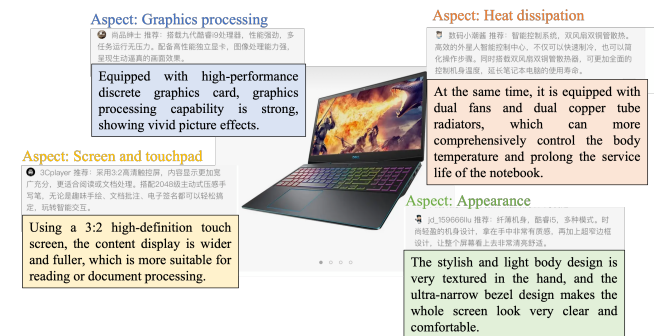

Figure 1: Examples of diverse product copywriting with various aspects: in the example of copywriting for computers, there are aspects related to appearances, graphics processing, screen, touchpad, heat dissipation, etc.

To address these issues, automatic product copywriting generation has become an essential line of research in e-commerce. Due to the great success of seq2seq models on natural language generation tasks, researchers adopt various neural architectures [19, 24–26] in this framework to utilize both the user-generated reviews and product information for product description/copywriting generation. These works provide the inspiration for designing the specific generators in generating product descriptions for e-commerce products [2, 4, 11, 15, 27, 29].

However, due to the diversity of e-commerce platform services, it is necessary to adapt the patterns of description writing to meet the different preferences of customers and sellers. As shown in Figure 1, various copywriting can show diverse aspects of the same product. Controlling the content aspects of the generated copywriting is significantly important for several reasons: (1) it helps attract different groups of customers with diverse copywriting contents. Different customers have different concerns about the same product. In other words, they will be attracted by most relevant aspect of the product based on their needs. For example, business people may care about the duration of battery power of a mobile phone while the game fans pay more attention to the screen refreshing rate; (2) it is flexible for sellers to display the most attractive aspects of the products, especially when the recommendation is combined with special promotion event or activity. For example, on Father's day or Mother's day, highlighting the "big screen for easy reading" and "concise interface" of mobile phones help promote the mobile phone as gifts for parents. Thus, controlling the aspects of the generated copyrighting is of great importance towards the various requirements from both the customers and platform sellers.

With the development of controllable text generation technique, controllable product description generation becomes possible. Chen et al. [2] explored to generate personalized product descriptions controllable by the customer preferences. Li et al. [13] and Liang et al. [14] presented an abstractive summarization system, where the summary can capture the most attractive aspects of a product. However, the methods mentioned above have several limitations for real-world controllable e-commerce product copywriting generation: (1) the generated copywriting from the generation models are likely to disrespect the truth of products, which is not acceptable to the real-world platform; (2) the aspects extracted by existing methods are based on adjective words in each sentence, without considering the sentence-level semantics; (3) existing clustering-based aspect assignment algorithm is hard to further involve more newly coming products for future model upgrading, which is impractical in many real-world scenarios; (4) there is no evaluation method to judge whether the generated copywriting matches the desired aspects.

To solve the aforementioned challenges and more importantly, to successfully implement and deploy the controllable product copywriting generator in the real world, large-scale e-commerce platform, we propose the E-Commerce Prefix-based Controllable Product Copywriting Generation (EPCCG). In this paper, we report our experience in developing the EPCCG system and deploying the EPCCG into the JD.com e-commerce product recommendation platform. The development of EPCCG consists of three main steps: 1) aspect extraction from copywriting, which is based on latent Dirichlet allocation (LDA) to extract aspects from copywriting corpus; (2) a novel phrase-based aspect classifier to label the copywriting as training data preparation; (3) product copywriting generation, which is built from a prefix-based e-commerce pre-trained model; and 4) knowledge-based post-processing, To the best of our knowledge, the proposed EPCCG is the first successful deployed controllable product copywriting generation system in the real-word e-commerce platform. Our contributions are summarized as follows:

(1) A domain-specific generative model EPCCG for product copywriting generation is proposed and deployed for the JD.com e-commerce product recommendation platform.
(2) We propose a phrase-based aspect classification method for automatically labelling the product copywriting with the extracted aspect. It can also be used for evaluating the aspect capturing ability of EPCCG.
(3) We extend the basic EPCCG model to Prompt-EPCCG by exploring various prompt strategies. The Prompt-EPCCG shows significant improvement towards the quality of generated product copywritng.
(4) We introduce the overall architecture of deployed system where the EPCCG is implemented into the large-scale JD.com recommendation platform. The experience learnt during deployment is also summarized.
(5) The experimental exploration results and the significant pay-off since deployment demonstrate the superiority of the proposed method over the baseline models and the effectiveness of its deployment in a real-world scenario.

2 RELATED WORK

Controllable text generation in Academic Exploration The typical technique based on neural network for text generation is the attention-based Seq2Seq model [19, 24–26, 28]. The attention-based Seq2Seq model has demonstrated its effectiveness in a number of tasks of text generation, including neural machine translation [24], abstractive text summarization [18], dialogue generation [21], etc. Controllable Text Generation (CTG) has recently attracted more focus from many researchers in the NLP community. The most relevant sub-topic in CTG to our problem is the Topic-based Generation[3, 9, 10]. CTRL [9] is an early attempt in this direction which trains a language model (i.e., Transformer) conditioned on a variety of control code including domain, style, topics, dates, entities, relationships between entities, etc. Though great efforts have been put into the exploration of controllable text generation, extending them into real-world product copywriting generation still remain large challenges, such as inconsistent attribute information before generated text and product information, and unavailability to the labeled copywriting datasets.

Product description generation Previous studies for text description generation in the e-commerce domain focused on statistical frameworks such as[6, 27, 29, 30], which incorporate statistical methods with the template for product descriptions generation. Such methods are limited by the hand-crafted templates. To this end, researchers adopted deep learning models and introduced diverse conditions into the generation model. Lipton et al. [15] generated reviews based on the conditions of semantic information and sentiment by language model. Khatri et al. [11] proposed a novel Document-Context based Seq2Seq models for abstractive and extractive summarizations in e-commerce. All aforementioned methods are for general product description generation without controlling the aspects of the generated contents.

Controllable Product description generation To meet the diverse pattern requirement of the generated product description, limited number of attempts have been made to control the generated product description regarding the length, sentiments and
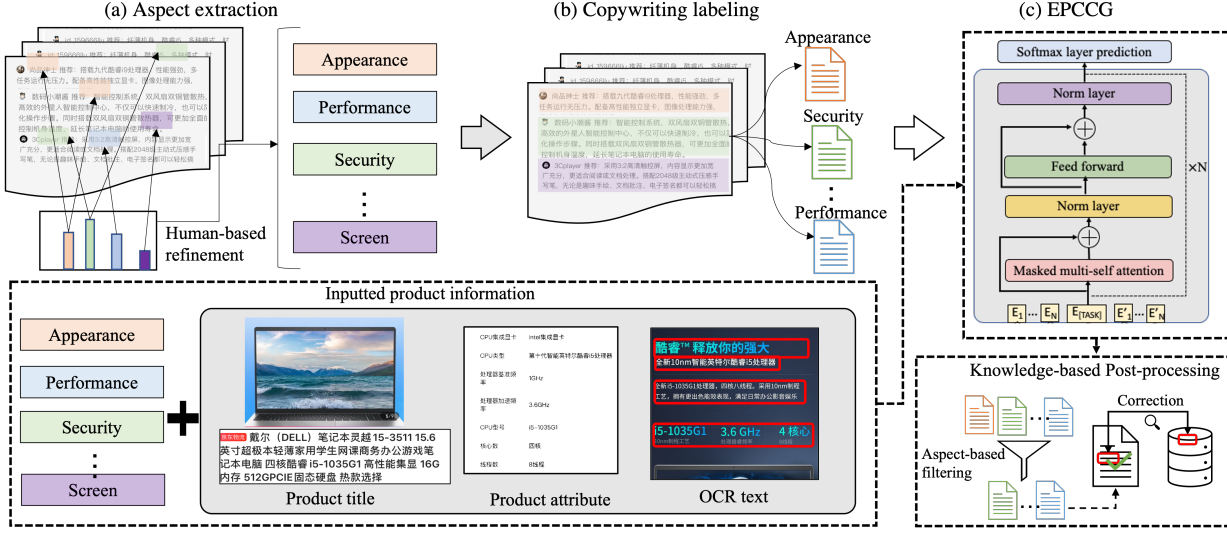
Figure 2: The development workflow of the proposed EPCCG: Training process: First, the aspects of a given kind of category of products are extracted (see sub-figure (a)), and the collected copywriting training samples are labeled with aspects (see sub-figure (b)); then the labeled copywriting are used for training the EPCCG (see sub-figure (c)). Inference/Generating process (circled in black dotted line): For each product, the product information and each desired aspect are combined as input, following which the knowledge-based post-process is conducted to guarantee the quality of the generated copywriting further.

topics. Chen et al. [2] explored to generate personalized product descriptions controllable by the customer preferences and adjective topics. Li et al. [13], Liang et al. [14] presented an abstractive summarization system that produces summary for Chinese e-commerce products, where the summary can capture the most attractive aspects of a product. However, these works all focus on introducing the single product generation model without matured deployment exploration in the large-scale e-commerce platform, where (1) an efficient and reusable aspect extraction and automatic data labeling process and (2) the precise consistent between copywriting and product information are required.

3 EPCCG: THE PROPOSED TECHNOLOGY

The development of EPCCG consists of three main parts, as illustrated in Figure 2. The first step is aspect extraction where the main aspects of each category of products are extracted from copywriting corpus. The second step is about copywriting labeling, namely, classifying the unlabeled copywriting samples based on the extracted aspects. In the third step, given the labeled copywriting, the E-commerce prefix-based controllable copywriting generator (EPCCG) is well trained and utilized. During production, the generated copywriting from EPCCG will be inputted into a correction and filtering module to improve their quality to meet the requirement of the industry e-commerce platform. The details of each main step are provided as follows:

3.1 LDA-based Aspect Extraction

Given the large volumes of product copywriting of a category of products, such as "skin care" or "mobile phones", aspect extraction aims to extract the key and popular topics from the copywriting

written by professional human writers. Here we summarize the sub-problem formulation.

Sub-problem formulation: aspect extraction. Given the collected copywriting datasets $\mathcal{D} = \{x_i\}_{i=1}^N$ from the human writers of a given product category, where N is the number of copywriting samples. The goal is to extract the key aspects $T = \{t_m\}_{m=1}^M$ of the given product category, where M denotes the number of aspects.

We proposed a LDA-based extractor for aspect extraction. The extractor contains two steps, as shown in Figure 2(a). First, the initial aspects are extracted by LDA model. Second, human-based refinement is conducted to balance the training samples and guarantee that the extracted aspects meet the e-commerce market preferences. We elaborate on each step as follows.

3.1.1 Topic Modeling via LDA. Topic Modeling in natural language documents aims to use unsupervised learning to extract the main topics in a collection of documents. LDA [8] is an unsupervised generative probabilistic method for modeling the semantic topic in corpus, which is the most commonly used topic modeling method. It assumes that each document can be represented as a probabilistic distribution over latent topics, and that topic distribution in all documents share a common Dirichlet prior. Each latent topic in the LDA model is also represented as a probabilistic distribution over words and the word distributions of topics share a common Dirichlet prior as well. Due to the space limit, more details of the typical LDA algorithm are provided in Appendix. D.

To select the best-performer hyper-parameters, we propose a dynamic-LDA inference-process. Specifically, we adopt the aspect coherence score to evaluate the performance of the LDA under each hyper-parameter setting. Aspect coherence measures the degree of semantic similarity between each pair of high-scoring words in

each aspect. We then select the LDA model which has the highest coherence score. Given the well-learned LDA model, we can select the aspects which have the most elevated aspect distribution probability. In this way, we can get the aspects of the given category of product $\mathcal{T} = \{t_m\}_{m=1}^M$ where each aspect contains several keywords.

3.1.2 Human-based refinement. Based on the keywords of each extracted aspect, we could easily interpret the semantic meaning of each aspect. However, for real-world industry application, the initial extracted aspects from the LDA model have several limitations: (1) the amount of training samples assigned for each aspect is not balanced, mainly influencing the following training process of the controllable text generation model; (2) some extracted aspects are not popular to attract customers, while some well-recognized aspects by professional marketing sellers are not discovered. Thus, we propose involving human-based refinement to adjust the extracted aspects slightly Based on the modified aspects and their semantics, it is easy to come up with the name for each aspect[1].

3.2 Unsupervised Phrase-based Aspect classification

A severe challenge of copywriting generation while real-world deployment is the lack of aspect-labeled copywriting. In this section, we propose the phrase-based aspect classifier, which can be utilized to assign the aspect to each product copywriting without human efforts. In addition, this well-trained aspect classifier can be utilized further to evaluate whether each generated copywriting match the desired aspect.

Sub-problem formulation: Extremely Weakly Supervised Aspect classification. Given the collected copywriting datasets $\mathcal{D} = \{x_i\}_{i=1}^N$ as well as the names of extracted key aspects $T = \{t_m\}_{m=1}^M$ of a given product category, where M denotes to the number of aspects and N denotes to the number of copywriting, the goal is to assign the aspect to each copywriting as $f : x_i \longrightarrow t_i$.

3.2.1 Limitations of existing LOTClass. We proposed a phrase-based aspect classifier inspired by the *LOTClass* [17] for aspect classification using only the names of aspects. The goal of *LOTClass* is only to use the label name of each class to train a classifier on unlabeled data without using any labeled documents. The main idea is to utilize the pre-trained language model for category name understanding to generate pseudo label for classification task and fine-tuning classifier on generated pseudo label data. However, the existing LOTClass has several limitations in dealing with the Chinese e-commerce product copywriting classification: (1) **Different language characteristic between Chinese and English**. Compared with English, Chinese single characters cannot carry out effective semantic information, and thus phrases are needed to better describe the semantic information of category names. (2) **Limited expressive ability in e-commerce domain**. The general phrase vocabulary cannot have limited coverage for the specific domain, e.g., e-commerce. Therefore, we have proposed the novel phrase-based LOTClass for e-commerce in Chinese to solve the above limitations.

---

[1]For brevity, we also use t_i to refer the name of each aspect.

3.2.2 The Overall workflow of phrase-based LOTClass. The phrase-based LOTClass consists of three main steps, as shown in Algorithm1. First and the most important, we need to train a domain specific phrase-based pre-trained language model $P(x)$. The details of training a phrase-based pre-trained language model is provided in the following Section 3.2.3. Second, after getting the e-commerce pre-trained language model, we rely on it to find a set of semantically similar substitute words for the name of the aspect To do this, in the corpus, e.g., copywriting dataset, for each aspect t_m, we mask the aspect and utilize the Phrase-based pre-training model to make predictions for these masked position. Normally, we select the top-K words based on their appearance frequency in the overall prediction to get the semantically similar substitute words set $\mathcal{S}^{(m)}$ for each aspect. At last, each copywriting is classified based on its coverage over the substitute words of the aspects. Specifically, as shown in Algorithm1, we calculate the coverage number $c_i^{(m)}$ of a given copywriting x_i regarding the substitute words set $\mathcal{S}^{(m)}$ of each aspect. The aspect that has the largest $c_i^{(m)}$ will be assigned to the copywriting x_i. As a result, all the copywriting will be labeled with an aspect.

Algorithm 1 Phrase-based LOTClass

Require: The collected copywriting datasets $\mathcal{D} = \{x_i\}_{i=1}^N$; The names of extracted key aspects $T = \{t_m\}_{m=1}^M$; The phrase-based pre-trained language model $P(x)$.

Ensure: The substitute words set $\mathcal{S}^{(m)}$ for each aspect t_m; The assigned aspect $y_i \in \mathcal{T}$ for each product copywriting x_i.

  for $m = 1 : M$ **do**

    Mask the aspect t_m in all of the copywriting of the corpus $\mathcal{D}$ to make the masked document $\bar{\mathcal{D}}$.

    Input the masked document $\bar{\mathcal{D}}$ to the pretrained language model $P(x)$ to predict the masked words.

    Based on the predicted words set $\bar{\mathcal{S}}^{(m)}$, select the Top-K words with the highest frequent appearances to form the substitute words set $\mathcal{S}^{(m)}$.

  end for

  for $i = 1 : N$ **do**

    for $m = 1 : M$ **do**

      calculate the coverage $c_i^{(m)}$ of x_i regarding the substitute words set $\mathcal{S}^{(m)}$

    end for

    Classify the copywriting x_i as $y_i = t_m$ where $c_i^{(m)}$ has the highest value.

  end for

3.2.3 Phrase-based Pre-trained Language Model. The core of the phrase-based aspect classifier is to learn a powerful e-commerce phrase-based pre-trained language model. There are two major limitations of the existing pre-trained model: (1) the basic unit in the vocabulary set of the existing pre-trained model is token (character in Chinese and word in English) and is hard to be used in tasks that need phrase-level knowledge; (2) training on the corpus in general domain made it be lack of domain-specific knowledge.

Thus, we propose to train a e-commerce phrase-based pre-trained language model.

Step 1: Building phrase-based Chinese e-commerce vocabulary. The proposed phrase-based Chinese e-commerce vocabulary consists of general Chinese characters and extracted domain-specific phrases. The general Chinese characters can be obtained any open-source basic Chinese character vocabulary. To extract the domain specific Chinese phrases, specifically, we first use part-of-speech tagging method [1] to extract high-frequency noun phrases from e-commerce corpus data as seed vocabulary. Since the number of these extracted phrases is limited and can not cover many insightful phrases. To make extension, the obtained seed vocabulary is used as the input of the phrase mining algorithm AutoPhrase [22] which are performed on the existing copywriting corpus to get the domain specific phrases.

Step 2: Phrase-based Chinese Pre-trained model. The next step is to tokenize the copywriting datasets based on the vocabulary. The principal of tokenization is to first find out all the phrases in the sentences which matches the phrase-based Chinese e-commerce vocabulary. The remaining sub-sentences will then be tokenized in a character-based fashion. The model architecture adopted here can be any language model.[2].

3.3 Prefix-based Controllable Copywriting Generation

In this section, we propose the E-commerce Prefix-based controllable copywriting generation model (EPCCG) and its extension, the Prompt-EPCCG. We first define the sub-problem for this step. Then, we introduce the preliminary of the basic prefix language model and the motivation to choose it. Next, we present the detailed techniques of the proposed EPCCG. Furthermore, we introduce the Prompt-EPCCG for the situation that only few labeled data can be used in training.

Sub-problem formulation: controllable copywriting generation Given product information text which is expressed in a sequences of the form $I = (w_1^I, ..., w_{L_1}^I)$ as well its relevant copywriting text $X = (w_1^X, ..., w_{L_2}^X)$, the goal is to learn a conditional language model $p_c(X)$ given the pre-defined aspect y_i as:

$$p_c(X) = \prod_{i=1}^{L_2} p(w_i^X | w_{<i}^X, I, y_i), \tag{1}$$

where aspect y_i provides a point of control over the generation.

3.3.1 Prefix Language Model. The prefix Language Model (LM) [5] is a left-to-right Language Model that decodes output sentences on the condition of a prefixed input sequence, which is encoded by the same model parameters with a bi-directional mask. The preliminary knowledge of prefix LM can be referred in Appendix C.

The flexible model structure of prefix LM makes it suitable for our application, where the power of language is utilized to realize our special training goals. Language can specify different parts of inputs and outputs as a sequence of symbols flexibly. Multiple kinds of information (i.e., product information and controller aspects) are embedded into the inputs represented by languages with specific tricks to build up the dataset. The language model will learn the

tasks implicitly. In this way, one model with the same structure can be reused for different tasks by only changing the input data which contains information of inputs, outputs and tasks all together.

3.3.2 EPCCG: The Proposed Model. To learn the conditional distribution in Eq 1 for E-commerce application, we proposed the E-commerce Prefix-based controllable copywriting generation model (EPCCG) based on the prefix LM. We utilize a 12-layer Transformer as our backbone network, where the input vectors are encoded to contextual representations through the Transformer blocks. Each layer of Transformer block contains multiple self-attention heads, which takes the output vectors of the previous layer as inputs.

Pre-training with E-commerce Corpus. During developing the model, our model sometimes generates descriptions with insufficient fluency or inaccurate information due to the limitation of training data. The model never reaches any similar related information for many new products before, making it hard to generate accurate product descriptions. To this end, we introduce domain-specific pre-training into the prefix LM. Instead of utilizing the general-domain pre-trained models obtained with large amounts of general knowledge and are not efficient enough for real-time online serving, we pre-train the domain-specific prefix-model with e-commerce knowledge collected from the JD.COM platform.
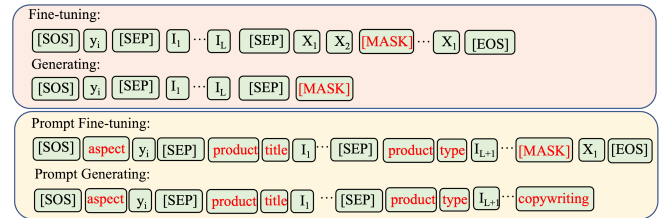

Figure 3: The text formulation for fine-tuning and generating process in EPCCG and Prompt-EPCCG

Fine-tuning in EPCCG. In the fine-tuning process, given the e-commerce pre-trained model, we pack the source sentence by concatenating the name of aspect y_i, the relevant product information I, and the target copywriting X together with special tokens as "[SOS]T[SEP]I[SEP]X[EOS]", as shown in Figure 3. Here the product information I includes product title, brand, attributes and OCR (optical character recognition) from product advertisement, where each word is initially represented as a vector, as shown in Figure 2. The model is fine-tuned by masking some percentage of tokens in the target sequence at random, and learning to recover the masked words. The training objective is to maximize the likelihood of masked tokens given context. [3]

Producing in EPCCG. In producing process, for each product, we pack the input sentence by combining the product information I with each desired aspect from set $\mathcal{T}$ that is extracted in Section 3.1. The input sentence is embedded and inputted into EPCCG to predict the remaining sequence as the generated copyrighting.

---

[2]For our deployment, we utilize the WoBERT[23] as the initial model.

[3][EOS], which marks the end of the target sequence, can also be masked during fine-tuning, thus when this happens, the model learns when to emit [EOS] to terminate the generation process.

3.3.3 Prompt-EPCCG. Recently, a new paradigm to utilize the pre-trained language model called "prompt-based learning" is proposed to address the issue of fine-tuning the pre-trained model on downstream tasks with few labeled data or no labeled data [16]. In this paradigm, instead of adapting pre-trained LMs to downstream tasks via objective engineering, downstream tasks are reformulated to look more like those solved during the original LM training with the help of a textual prompt. In other words, prompt-based learning is a more efficient way to utilize the knowledge in the pre-trained model with fewer labeled data. Thus, we proposed the "Prompt-EPCCG" to enhance the performance of EPCCG on personalized copywriting generation when only very few labeled data could be used for training.

Specifically, we adopt the Fixed-prompt LM Tuning strategy [20] and reorganize the input sequence as

$$[SOS]aspect : T[SEP]title : I_1[SEP]brand : I_2[SEP]attribute :$$
$$I_3[SEP]type : I_4[SEP]OCR : I_5[SEP]copywriting : X[EOS]$$

which means there will be an additional prefix before each detailed product information to help LM better understand the input. We also explore the performance of other possible prompt designs and the experiment results are shown in Section 5.

3.4 Knowledge-based Post-processing

At the early stage of development, the generated copywriting still have some issues to be directly displayed into the real-world platform: (1) some mentioned attributes of product in the generated copywriting are not consistent with the real product information; (2) the contents of generated copywriting do not match the desired aspect. To overcome these issues, we introduce the knowledge-based post-process to ensure the quality of the generated copywriting during real-world deployment. The knowledge-based post-process consists of two main steps: attribute-based correction and aspect-based filtering.

To solve the first issue, we pre-define a list of Regular Expressions (RE) to extract the values of the common attributes from the copywriting. We can get the correct attribute value from our knowledge base. Then we compare the two values and replace the value in copywriting with the correct one. To address the second issue, we resort to the phrase-based aspect classifier mentioned in Section 3.2. If the predicted aspect of the generated copywriting does not match the desired one, it will be filtered.

4 SYSTEM DEPLOYMENT

In this section, we introduce the experience in how to deploy the proposed EPCCG into our online e-commerce recommendation platform. As shown in Figure 4, the overall system consists of the following components:

- Daily model training: This module collects data, including product information and descriptions, to train the EPCCG model. Each collected product copywriting is assigned an aspect by the phrase-based aspect classifier, as shown in Figure 4 (a).
- EPCCG daily production: New product information will be collected and combined with each aspect, which will then be inputted into the EPCCG for copywriting generation.
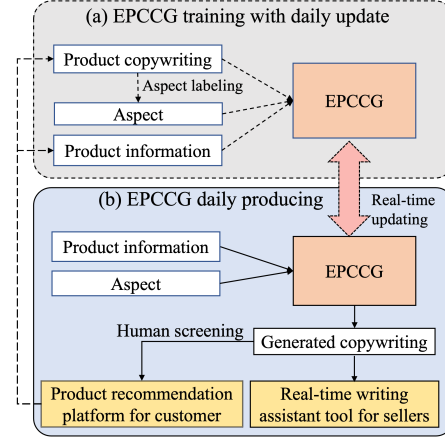

Figure 4: Deployment of EPCCG in the E-Commerce recommendation platform

After the human screening, the contents produced here are directly used by both the product recommendation platform for JD.com customers.
- Real-time writing assistant tool for sellers: A real-time writing assistant tool based on updated EPCCG has also been utilized for JD sellers. With this tool, JD sellers can input their desired products, and obtain product descriptions automatically. Sellers can then edit or select the generated contents, and display them to customers.
- The automatic procedure with daily update: EPCCG is trained daily to meet the latest writing preference based newly updated data collected from the recommendation platforms.

5 EXPERIMENT AND PAYOFF

In this section, we first introduce the dataset, the compared models and the evaluation metrics. We then demonstrate the experimental results in a series of evaluations and further analyze the development selections. In addition, the real industry payoff of the proposed EPCCG system after deployment and several practical cases are provided. Details of implementation are provided in Appendix A.

Table 1: The extracted aspects of different product categories

Dataset	Extracted aspects
computer	appearance,screen, graphic processing battery,security, heat dissipation, keyboard capability, camera
men clothes	fabric,version, pattern, color, match, style, pocket, collar
home appliances	performance, degerming, capability, noises intelligent control, energy conservation appearance

5.1 Dataset

Considering that there is a lack of large-scale open-source datasets for this task, we constructed a new dataset *JDCopywriting*, containing the basic information of the products, including title, OCR

Table 2: Machine-based and human-based evaluation results on mobile datasets.

methods	rouge-1	rouge-2	rouge-L	bleu-1	bleu-2	bleu-3	bleu-4	meteor	sacrebleu	aspect	validness
CTRL	0.2056	0.0954	0.1723	0.1637	0.1109	0.0806	0.0627	0.1603	3.9511	91.05%	93.88%
conditional GPT	0.2012	0.0667	0.1917	**0.2247**	0.1305	0.0935	**0.0938**	0.1338	5.0886	95.55%	95.66%
EPCCG w/o pretraining	0.2023	0.0635	0.1875	0.1886	0.1013	0.1009	0.0947	0.1394	5.2836	95.00%	95.80%
discrete-code based EPCCG	0.2228	0.1072	0.1877	0.1770	0.1235	0.0918	0.0725	0.1680	4.6739	94.32%	96.25%
label-code based EPCCG	0.2190	0.1073	0.1836	0.1669	0.1179	0.0883	0.0699	0.1728	4.7519	93.02%	96.10%
name-code based EPCCG	**0.2230**	**0.1234**	**0.2105**	0.1909	**0.1359**	**0.1027**	0.0821	**0.1770**	**5.9966**	**95.71**%	**96.29%**

text, attribute, as well as the copywriting of products with labeled aspects. The data are collected from JD.COM platform, a large-scale website for e-commerce in China. The product information and the copywriting are composed by the sellers and content writers on the website from Jan 2020 to June 2021. Each data instance is automatically annotated with its aspect based on the proposed phrase-based aspect classifier introduced in Section 3.2.

The whole dataset covers six categories of products, including *men clothes*, *mobile-phone*, *computers*, *women clothes*, *facial skin care*, and *home appliances*, each of which covers 500, 000 data samples in total. Each of them are extracted with several aspects, as illustrated in Table 1. The aspects of more categories are provided in Appendix D.

5.2 System for Comparison

In this section, we introduce the baseline and choices for our model components while developing EPCCG and the prompt-EPCCG.

5.2.1 Baseline of EPCCG. To validate the superiority of the proposed EPCCG, we compare it with two SOTA controllable text generation models:

(1) **Conditional Transformer Language Model(CTLR)** [9] which is a conditional transformer based LM, trained to condition on control codes that govern style, content, and task-specific behavior.

(2) **Conditional GPT** [12] which is based on OpenAI GPT-2 to generate abstractive and comprehensive information based on keywords extracted from the original articles.

5.2.2 Choices of Controllable Code Pattern. To explore the best way to involve the controllable aspect in the input sentence, we test three settings:

(1) **Discrete-code based** treats the controllable aspect as a category label by a one-hot vector. The one-hot vector is then concatenated with the input product information.

(2) **Label-code based** also treats the controllable aspect as a category label but tokenizes it as a special token.

(3) **Name-code based** directly concatenates the name of the aspect with the product information for tokenization.

5.2.3 Choices of Prompt Design. The key point to a successful prompt-model is the design of the prompt paradigm. To explore the best prompt paradigm, we compare the five typical settings.

(1) **prompt-EPCCG w/o sep**, where the prefix is added in a simple way such as "'*aspect:...product:...copywriting:...*" to format the input.

(2) **prompt-EPCCG with sep**, where we add the prefix with "'*[SEP]*" token as "'*aspect:....[SEP] product:....[SEP] copywriting:...*" to format the input.

(3) **prompt-EPCCG-advance w/o sep**, where we further separate the input with more detailed prefix as "'*aspect:...product title:...product type:...product attribute:...copywriting:...*".

(4) **prompt-EPCCG-advance with sep** add "'*[SEP]*" onto the input of **prompt-EPCCG-advance w/o sep**.

5.3 Evaluation Metrics

We evaluate our model on generation quality and aspect capturing ability based on human and machine evaluation metrics.

Machine-based Evaluation for Copywriting Quality. A series of typical BLEU (i.e., bleu-1, bleu-2, bleu-3, bleu-4 and sacrebleu) as well as ROUGE scores (i.e., rouge-1, rouge-2, rouge-L and meteor) are utilized to evaluate the similarity between the generated and ground-truth copywriting regarding the N-gram cases.

Machine-based Evaluation for Aspect Capturing Ability. To judge whether the generated product copywriting matches the desired controllable aspect, we utilize the phrase-based aspect classifier[4] to classify the aspect of the generated copywriting. We calculate the percentage of the correctly matched copywriting among the total generation, mentioned as *aspect*.

Human-based Evaluation. We also depend on human reviewers to judge whether a produced copywriting is valid and informative for online display, mentioned as *validness*. It is worth noting that this evaluation is also an online metric since deployment.

5.4 Experiment Results and Analysis

In this section, we analyze the experimental results by focusing on a few issues illustrated as below. We utilize the *men clothes* category as an example to explore the answers to the issues.

Does different pattern of control code influence the performance of copywriting generation? We compared three different patterns of control aspect involvement as mentioned in Section 5.2.2. The experiment results on *men clothes* category are provided in Table 2. Based on the results, concatenating the name of aspect with the input text and tokenizing it in the same way as the input product information achieved the best performance. Specifically, the *name-code based EPCCG* outperforms the other two patterns with an advantage of +1.34 sacreBLEU (relatively 21.2%). The highest *aspect* of *name-code based EPCCG* validates that involving the

---

[4]The well-trained classifier is obtained and stored during the data labeling process as introduced in Section 3.2.

Table 3: Machine-based and human-based evaluation results on different prompt setting for EPCCG.

datasests	rouge-1	rouge-2	rouge-L	bleu-1	bleu-2	bleu-3	bleu-4	meteor	sacrebleu	aspect
prompt-EPCCG w/o sep	0.2282	0.0755	0.2114	0.3004	0.2085	0.1492	0.1131	0.1437	8.2289	96.77%
prompt-EPCCG with sep	0.2293	0.0755	0.2125	**0.3019**	**0.2092**	0.1492	0.1133	**0.1444**	8.3290	97.22%
prompt-EPCCG-advance w/o sep	0.2275	0.0754	0.2107	0.2959	0.2056	0.1472	0.1114	0.1422	8.2096	97.02%
prompt-EPCCG-advance with sep	**0.2317**	**0.0791**	**0.2157**	0.2966	0.2072	**0.1495**	**0.1141**	0.1432	**8.51**77	**97.42%**

Table 4: The produced copywriting generated by varying the aspect attribute while fixing the product title as input

category	aspects	produced copywriting
computers	"appearances"	"This notebook uses a thin and light body design, easy to carry, more handy when going out to work"
	"heat dissipation"	"Built-in high-density thin fan blades and double heat pipe design, the heat is discharged through the whole cooling system".
	"battery"	"The built-in large-capacity battery has a long battery life, helping you to bid farewell to the power crisis calmly".
home appliances	"performance"	"Equipped with D-type evaporator to increase the cooling area and improve the cooling performance".
	"noises"	"Using Fisher-Paykel direct drive variable frequency motor, the operation is smooth and smooth, and the noise is effectively reduced".
	"energy conservation"	"First-class energy efficiency, providing energy saving and electricity saving".

semantic meaning of aspect help improve the aspect capturing capability of the text generation model. Thus, we finally select the *name-code based EPCCG* as the final implemented pattern.

Does the proposed EPCCG architecture bring the best performance for copywriting generation? We compared the proposed EPCCG with the state-of-the-art controllable text generation model as mentioned in Section 5.2.1. As shown in Table 2, the *name-code based EPCCG* achieves a higher score in seven ROUGE and BLEU metrics over CTRL and conditional GPT. Whichever controllable code pattern is selected, the proposed EPCCG have the highest validness by human evaluation in the real-world industrial situation over CTRL and conditional GPT by 2.42% and 0.63%.

Is the proposed prefix-based pre-training strategy necessary for the improvement of performance? We conduct an ablation study by comparing the performance of the proposed EPCCG with and without the pre-training strategy. As shown in Table 2, given the *name-code based EPCCG* and *EPCCG w/o pre-training* setting which have the same architecture and controllable code pattern, *name-code based EPCCG* significantly outperforms *EPCCG w/o pre-training* in almost all the evaluation metrics, such as an advantage of +0.5 *aspect*. Specially, adding domain-specific pre-training strategy improved the *meteor* by around 4.32.

Does the design of the prompt paradigm influences the quality of the generated copywriting? As shown in Table 3, we compare the different designs of prompt paradigm mentioned in Section 5.2.3 based on the *men clothes* category. Overall, the setting *prompt-EPCCG with sep* achieves the best performance regarding both the quality of generated copywriting as well as the capturing ability of aspect. Based on the results that *prompt-EPCCG with sep* outperforms *prompt-EPCCG w/p sep* with advantage of +0.45

aspects and *prompt-EPCCG-advance with sep* outperforms *prompt-EPCCG-advance w/o sep* with advantage of +0.38 aspects, it can be conclude that adding the token of "[SEP]" improves the aspect capturing ability. Based on the results that *prompt-EPCCG-advance with sep* outperforms *prompt-EPCCG with sep* with advantage of +0.2 aspects and +0.2 sacreBLEU, it can be concluded that formatting more structured product information text can help the model better understand the semantic meaning of input.

5.5 Case Studies

In this section, we perform case studies to observe how the proposed EPCCG influenced the generation so that the model can generate different contents based on various controllable aspects. Table 4 presents the generated examples by the models with varying aspects for home appliances and computers (have been translated into English). The generated copywriting can talk about the product's different aspects by successfully capturing the desired.

5.6 Payoff After Deployment

EPCCG has been deployed in the JD.com product recommendation platform since October 2021. Since deployment, EPCCG has covered 6 main categories of products, including women clothes, men clothes, mobile-phone, computers, appliances, and facial care, by generating and delivering **56,000** copywriting and promoting **millions of Gross Merchandise Volume**. The deployed EPCCG system serves for the content generation for customers. Figure 5 shows an example of the product recommendation platform for customers, "Discovery Goods Channel". The English translations of the generated texts are shown in the pop up bubbles in the figure. The generated product descriptions, paired with a short title, are pushed to the Discovery Goods Channel in the JD e-commerce website.

Figure 5: Examples of EPCCG generated descriptions on the JD e-commerce product recommendation platform.

6 CONCLUSIONS

In this paper, we introduced a deployed controllable product copywriting generation system in a large-scale e-commerce platform. We proposed a novel model EPCCG and its extension prompt-EPCCG based on the prefix-based pre-trained language model. Our extensive experiments showed that our method outperformed the baseline models through various evaluation metrics, including machine-based and human evaluations. We have successfully deployed the proposed framework onto JD.com, one of the world's largest online e-commerce platforms, with significantly payoff. A large volume dataset for product description generation, namely JDCopywriting, has been generated and annotated, which can be used as a benchmark dataset for future research in this field.

REFERENCES

[1] Eric Brill. 1992. *A simple rule-based part of speech tagger*. Technical Report. PENNSYLVANIA UNIV PHILADELPHIA DEPT OF COMPUTER AND INFORMATION SCIENCE.

[2] Qibin Chen, Junyang Lin, Yichang Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Towards knowledge-based personalized product description generation in e-commerce. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3040–3050.

[3] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *International Conference on Learning Representations*.

[4] Vijay Daultani, Lasguido Nio, and Young-Joo Chung. 2019. Unsupervised extractive summarization for product description using coverage maximization with attribute concept. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. IEEE, 114–117.

[5] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pretraining for natural language understanding and generation. *Advances in Neural Information Processing Systems* 32 (2019).

[6] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1602–1613.

[7] Xiaojie Guo, Shugen Wang, Hanqing Zhao, Shiliang Diao, Jiajia Chen, Zhuoye Ding, Zhen He, Yun Xiao, Bo Long, Han Yu, et al. 2021. Intelligent Online

[8] Selling Point Extraction for E-Commerce Recommendation. *arXiv preprint arXiv:2112.10613* (2021).

[8] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. 2019. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications* 78, 11 (2019), 15169–15211.

[9] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* (2019).

[10] Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2020. A Distributional Approach to Controlled Text Generation. In *International Conference on Learning Representations*.

[11] Chandra Khatri, Gyanit Singh, and Nish Parikh. 2018. Abstractive and extractive text summarization using document context vector and recurrent neural networks. *arXiv preprint arXiv:1807.08000* (2018).

[12] Virapat Kieuvongngam, Bowen Tan, and Yiming Niu. 2020. Automatic text summarization of covid-19 medical research articles using bert and gpt-2. *arXiv preprint arXiv:2006.01997* (2020).

[13] Haoran Li, Peng Yuan, Song Xu, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Aspect-aware multimodal summarization for chinese e-commerce products. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8188–8195.

[14] Jiahui Liang, Junwei Bao, Yifan Wang, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2021. CUSTOM: Aspect-Oriented Product Summarization for EC ommerce. In *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 124–136.

[15] Zachary C Lipton, Sharad Vikram, and Julian McAuley. [n. d.]. Capturing meaning in product reviews with character-level generative text models. ([n. d.]).

[16] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586* (2021).

[17] Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. *arXiv preprint arXiv:2010.07245* (2020).

[18] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 280–290.

[19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.

[20] Timo Schick and Hinrich Schütze. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 255–269.

[21] Iulian Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron Courville. 2017. Multiresolution recurrent neural networks: An application to dialogue response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[22] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1825–1837.

[23] Jianlin Su. 2020. *WoBERT: Word-based Chinese BERT model - ZhuiyiAI*. Technical Report. https://github.com/ZhuiyiTechnology/WoBERT

[24] 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.

[25] 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.

[26] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *arXiv preprint arXiv:1506.03134* (2015).

[27] Jinpeng Wang, Yutai Hou, Jing Liu, Yunbo Cao, and Chin-Yew Lin. 2017. A statistical framework for product description generation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 187–192.

[28] Lingfei Wu, Ian En-Hsu Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J Witbrock. 2018. Word Mover's Embedding: From Word2Vec to Document Embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4524–4534.

[29] Joan Xiao and Robert Munro. 2019. Text Summarization of Product Titles.. In *eCOM@ SIGIR*.

[30] Xueying Zhang, Yanyan Zou, Hainan Zhang, Jing Zhou, Shiliang Diao, Jiajia Chen, Zhuoye Ding, Zhen He, Xueqi He, Yun Xiao, et al. 2021. Automatic Product Copywriting for E-Commerce. *arXiv preprint arXiv:2112.11915* (2021).

A HYPER-PARAMETERS AND ARCHITECTURE FOR REPRODUCING

A.1 The hyper-parameters of phrase-based aspect classifier

In this section, we introduce the detailed setting of the proposed phrase-based aspect classifier. To select the substitute words for each aspect, we select the Top-50 words which with the highest frequent appearances, namely we let $K = 50$. Next, the core components of the proposed classifier is the pre-trained language model. The pre-trained model is based on BERT model which consists of 12 transformer layers. While training, the learning rate is 0.00003 and the batch size is 220. The detailed hyper-parameters for architecture are listed in Table 5.

Table 5: The detailed hyper-paramters of arechtecture of pre-trained LM in aspect classifier.

hyper-parameters	value
attention_probs_dropout_prob	0.1
embedding_size	768
hidden_act	"gelu"
hidden_dropout_prob	0.1
hidden_size	192
initializer_range	0.02
intermediate_size	768
layer_norm_eps	1e-12
max_position_embeddings	512
num_attention_heads	3
num_hidden_layers	12
output_past	true
pad_token_id	0
summary_activation	"gelu"
summary_last_dropout	0.1
type_vocab_size	2
vocab_size	21128

A.2 The hyper-parameters of EPCCG model

In this section, we introduce the hyper-parameters of the architecture of prefix-based controllable copywriting generation model while implementation. The backbone network is a 12-layer transformer with multi-head attentions. While training, the learning rate is 0.00003 and the batch size is 220. The detailed hyper-parameters for architecture are listed in Table 6.

B PRELIMINARY OF LDA

Topic Modelling in natural language document aims to use unsupervised learning to extract the main topics (represented as a set of words) that occur in a collection of documents. LDA [8] is an unsupervised generative probabilistic method for modeling a corpus, which is the most commonly used topic modeling method. It assumes that each document can be represented as a probabilistic distribution over latent topics, and that topic distribution in all documents share a common Dirichlet prior. Each latent topic in

Table 6: The detailed hyper-paramters of architecture of pre-trained LM in aspect classifier.

hyper-parameters	value
attention_probs_dropout_prob	0.1
embedding_size	768
hidden_act	"gelu"
hidden_dropout_prob	0.1
hidden_size	192
initializer_range	0.02
intermediate_size	768
layer_norm_eps	1e-12
max_position_embeddings	512
num_attention_heads	12
num_hidden_layers	12
output_past	true
pad_token_id	0
summary_activation	"gelu"
summary_last_dropout	0.1
type_vocab_size	2
vocab_size	21128

the LDA model is also represented as a probabilistic distribution over words and the word distributions of topics share a common Dirichlet prior as well. In the process of product aspect extraction, given the copywriting dataset $\mathcal{D} = \{x_i\}_{i=1}^N$ and each copywriting x_i contains $H^{(i)}$ words $w_k^{(i)}$, the whole copywriting corpus $\mathcal{D}$ is modeled in the following process:

- Choose a distribution $\phi \backsim Dir(\beta)$ for aspect.
- Choose a distribution $\theta \backsim Dir(\alpha)$ for copywriting.
- For each word $w_k^{(i)}$ in copywriting x_i:
  - (a) Choose a topic $t_m^{(i)} \backsim Multinomial(\phi)$.
  - (b) Choose a word $w_k^{(i)} \backsim Multinomial(\theta)$.

Thus, to learn this generative mode, the goal is to maximize the overall objective as:

$$\max_{\phi,\theta} \prod_{i=1}^N \int p(\theta^{(i)}|\alpha)(\prod_{k=1}^{H^{(i)}} \sum_{t_m^{(i)}} p(z_m^{(i)}|\theta^{(i)})p(w_k^{(i)}|t_m^{(i)},\beta)))d\theta^{(i)}$$

(2)

In the process of product aspect extraction, given the copywriting dataset $\mathcal{D} = \{x_i\}_{i=1}^N$ and each copywriting x_i contains $H^{(i)}$ words $w_k^{(i)}$. where α, β and M are hyper-parameters about LDA. Here We use the Gibbs sampling, which is a Monte Carlo Markov-chain algorithm to estimate the LDA parameters. After this, we can successfully identify the main aspects from the copywriting corpus.

C PRELIMINARY: PREFIX LANGUAGE MODEL

The prefix Language Model (LM) [5] is a left-to-right Language Model that decodes output sentence on condition of a prefixed input sequence, which is encoded by the same model parameters with a bi-directional mask. As shown in Figure 4, during training process, the input tokens can attend to each other bidirectionally, while the output tokens can only attend to tokens on the left. To realize this,

Table 7: Machine-based and human-based evaluation results on different categories of aspects based EPCCG.

datasests	rouge-1	rouge-2	rouge-L	bleu-1	bleu-2	bleu-3	bleu-4	meteor	sacrebleu	aspect	validness
mobile phone	0.2054	0.0674	0.1904	0.2956	0.2015	0.1432	0.1082	0.1386	7.6870	96.02%	96.29%
women clothes	0.1770	0.0454	0.1641	0.2584	0.1669	0.126	0.0938	0.1338	5.0886	95.17%	91.40%
facial skin care	0.1863	0.0468	0.1667	0.2581	0.1702	0.1138	0.0810	0.1429	5.5697	90.50%	96.25%
home appliances	0.1928	0.0606	0.1825	0.2660	0.1804	0.1262	0.0936	0.1266	6.2596	97.10%	99.45%
computer	0.1969	0.0653	0.1811	0.2112	0.1513	0.1110	0.0849	0.1684	7.1225	96.16%	95.06%

Table 8: The extracted aspects for different categories of products.

Dataset	Extracted aspects
Mobile-phone	appearance,screen, network,camera battery,security,capability
computer	appearance,screen, graphic processing battery,security, heat dissipation, keyboard capability, camera
men clothes	fabric,version, pattern, color, match, style, pocket, collar
women clothes	fabric,version, pattern, color, match, style, pocket, collar
facial skin care	sun-screening, brightening, moisturizing, cleaning, anti-wrinkling, package design repair and soothing, ingredient
home appliances	performance, degerming, capability, noises intelligent control, energy conservation appearance

a corrupted text reconstruction objective is usually applied over input tokens, and an auto regressive language modeling objective is applied over output tokens, which encourages the prefix LM to better learn representations of the input. In this way, the prefix LM can model the generative process of $p(w_i^X | w_{<i}^X)$ for each sentences.

Specifically, the input vectors is first packed as $H^0 = [w_1^I, ..., w_{L_1}^I]$, and then encoded into contextual representations at different levels of abstract $H^l = [h_t^l, h_1^l, h_n^l]$ using an L-layer Transformer $H^l = Transformer^l(H^{l1}), l \in [1, L]$. In each Transformer block, multiple

self-attention heads are used to aggregate the output vectors of the previous layer. For the $l - th$ Transformer layer, the output of a self-attention head A^l is computed via:

$$Q = H^{l-1}W_Q^l, K = H^{l-1}W_K^l, V = H^{l-1}W_V^l \qquad (3)$$

$$M_{ij} = \begin{cases} 0 & allow\ to\ attend \\ -\infty & prevent\ from\ attending \end{cases} \qquad (4)$$

$$A_l = softmax(\frac{QK^\top}{\sqrt{d_k}}) + MV_l \qquad (5)$$

where the previous layer's output $H^{l1} \in R^{n \times d_h}$ is linearly projected to a triple of queries, keys and values using parameter matrices W_Q^l, W_K^l, $W_V^l R^{d_h \times d_k}$, respectively, and the mask matrix $M \in R^{n \times n}$ determines whether a pair of tokens can be attended to each other.

D THE EXTRACTED ASPECTS FOR DIFFERENT CATEGORIES OF PRODUCTS

In this section, we provide the extracted aspects for all the six categories of products in the JDCopywriting dataset.

E EVALUATION RESULTS ON DIFFERENT DATASETS

As shown in Table7, the proposed EPCCG shown effectiveness in the other five categories of products. Specifically, the proposed EPCCG performs the best while dealing with the category of *home appliance*. This is because the *home appliance* has the most standard attributes information which make the input text much more structured. Following *home appliance* are *mobile phones* and *computers*, which are all belong to the electrical line and have the most standard attribute information, resulting the standard aspects.