Self-supervised Product Title Rewrite for Product Listing Ads

Xue Zhao, Dayiheng Liu, Junwei Ding, Liang Yao,

Yao Yan, Huibo Wang, Wenqing Yao
Alibaba Group

{xue.zx,liudayiheng.ldyh,djw99219}@alibaba-inc.com
{yaoliang.yl,yanyao.yy}@alibaba-inc.com
{huibo.whb,wenqing.ywq}@alibaba-inc.com

Abstract

Product Listing Ads (PLAs) are primary online advertisements merchants pay to attract more customers. However, merchants prefer to stack various attributes to the title and neglect the fluency and information priority. These seller-created titles are not suitable for PLAs as they fail to highlight the core information in the visible part in PLAs titles. In this work, we present a title rewrite solution. Specifically, we train a self-supervised language model to generate high-quality titles in terms of fluency and information priority. Extensive offline test and real-world online test have demonstrated that our solution is effective in reducing the cost and gaining more profit as it lowers our CPC[1], CPB[2] while improving CTR[3] in the online test by a large margin. It is also easy to train and deploy, which can be a best practice of title optimization for PLAs.

1 Introduction

Product Listing Ads (PLAs) are crucial online marketing tools for merchants to attract more customers and encourage them to click their ads. They have different names in various ads channels, for example, Dynamic Product Ads in Facebook and Instagram, Shopping Ads on Google, as shown in Fig 1. PLAs usually have a limit on display text length, for instance, in Google Shopping Ads, users can see only the first 70 or fewer characters of the title[4]). Therefore, PLAs titles are expected to reveal the product type and core attributes earlier so that users can clearly identify the product, as illustrated in Table 1. However, to trigger ads more often and affect the user's purchase intention more positively, sellers list as many attributes as possible in the title without considering the fluency and readability, most importantly, the information priority. as illustrated in Table 2. These titles fail to highlight the core information and make it difficult to comprehend as a whole.

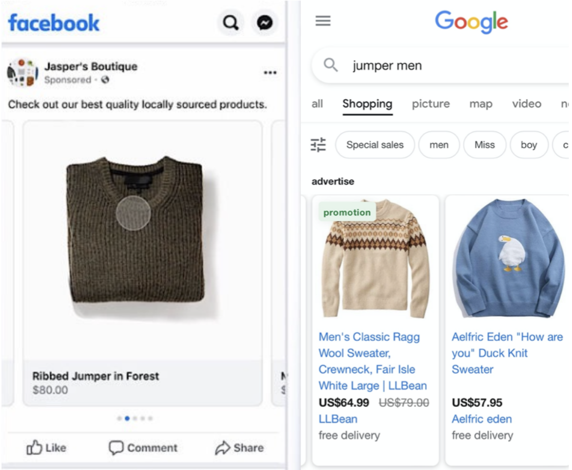

Figure 1: Product Listing Ads from different channels

Existing work has made the attempt to generate titles from keywords(de Souza et al., 2018) and product images(Zhang et al., 2019), or generate description text(Shao et al., 2021) for products, however, little work has investigated the title optimization for PLAs. At first, we explored the rule-based method by assigning weight to attribute words and reordering the words/chunks by weight. However, the rule-based method heavily relies on the accuracy of attribute detection, phrase boundary detection, and the appropriateness of attribute weights. It is hard to optimize rules without exhausting human effort. Therefore, we attempt to use language models in our title rewrite task.

The biggest obstacle of model-based method is the lack of high-quality titles regarding fluency and information order as labels for supervised learning. In this work, we solve the problem by performing self-supervised learning. Instead of writing high-quality titles as labels, we design a multi-level shuf-

[1] Cost Per Click

[2] Cost Per Buyer

[3] Click Through Rate

[4] https://support.google.com/merchants/answer/6324415?hl=en

Proceedings of NAACL-HLT 2022: Industry Track Papers, pages 79 - 85
July 10-15, 2022 ©2022 Association for Computational Linguistics

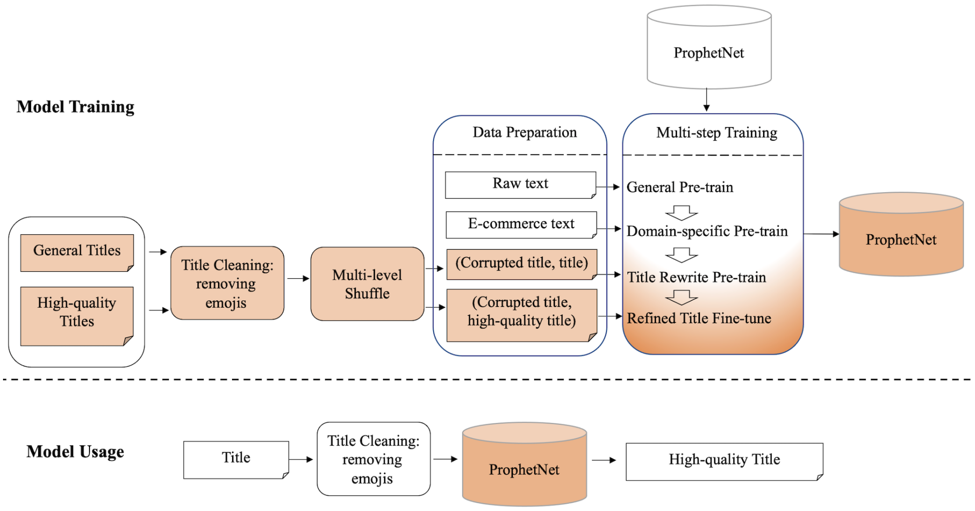

Figure 2: The framework of our product title rewrite solution. The upper part shows model training details: title cleaning, multi-level shuffling, data preparation, and multi-step training. The multi-level shuffle module creates (pseudo title, title) pairs for self-supervised training. The lower part illustrates the model usage: the input title is cleaned and then input to the trained ProphetNet to generate a high-quality one.

Priority	**Attribute Type**	**Attribute Value**
1st	Product Type	phone case
2nd	Core	with magsafe, for iphone13, leather
3rd	Common	new style, golden brown, 6.1 inches

Quality	**Title Example**
good	iphone13 leather phone case with magsafe new style golden brown 6.1 inches
bad fluency	with magsafe new style golden brown 6.1 inches phone case for iphone13 leather
bad priority	new style golden brown 6.1 inches phone case for iphone13 leather with magsafe

Table 1: Examples of attribute priority for a phone case and possible titles. Small case is used in the paper.

title	suitable for apple 12pro mobile phone case iphone12 protective case genuine leather drop-resistant new style all-inclusive silicone ultra-thin 11pro max high-end for men and women limited
optimized	mobile phone case silicone ultra-thin genuine leather protective case drop-resistant suitable for apple 12pro iphone12 11pro max all-inclusive new style high-end for men and women limited.

Table 2: Example of product title optimization.

fle module that uses titles to generate low-quality pseudo titles. Then the language model is trained on (pseudo title, title) pairs, during which it can learn to reorder the words to recover the original titles. Moreover, we propose a multi-step training procedure consisting of pre-train and fine-tune to enable the model to generate good titles.

The overall framework of our solution is illustrated in Fig 2. Moreover, for the sake of information accuracy, we only focus on ***information reordering*** and avoid any word insertion, deletion, and modification to the original title.

2 Method

We first introduce our multi-level shuffle module, which creates the (pseudo title, title) pairs for self-supervised training. Then we elaborate on the multi-step training procedure. It is worth noting that our solution can be built upon any language models, such as BART (Lewis et al., 2019) and GPT2 (Radford et al., 2019). We use the ProphetNet (Qi et al., 2020) framework in practice as it is superior to BART and GPT2 and has achieved new state-of-the-art in multiple text generation tasks(Dayiheng Liu and Duan, 2020).

2.1 Multi-level Shuffle Module

We overcome the absence of a learning target by thinking about the problem from an interesting perspective. The only available titles are seller-created: accurate, informative while uneven in quality, where good titles can train the model to generate better while the bad ones can also be good learning targets in terms of wording and phrasing of attributes, grammar, and the semantic context of the words. If we shuffle the word order of titles as input, even the bad titles become good supervision as they have better fluency than the corrupted ones. In light of these considerations, we build a multi-level shuffle module to mimic the problematic titles and generate low-quality pseudo titles as model input. Specifically, we have three strategies to cover almost all the word order issues in the titles.

Chunk-level We use the chunking tools[5] to split the title into chunks, then we randomly swap two or more chunks to obtain low-quality titles. From Table 1, the good title can be split into *"iphone13 leather phone case | with magsafe | new style | golden brown | 6.1 inches"*. After shuffling, the title may become the bad ones in Table 1.

Span-level We create the text spans by combining the random number of adjacent chunks arbitrarily into a larger text span without overlapping, then we randomly exchange the position of two or more spans. This strategy generates the easiest case for the model to learn because most of the words are still in proper order after shuffling.

Token-level After tokenization, the title is split into a list of tokens. We switch the position of two or more tokens to mimic the word order issue with the highest severity since it needs a more complicated adjustment to recover.

In practice, we make sure to keep 15% titles unchanged. We apply chunk-level strategy to 55% titles, span-level strategy to 25% titles, and process only the rest 5% titles with token-level strategy because such messy corruption hardly happens in titles while a large portion of such hard cases will delay the model convergence.

2.2 Model Training

We introduce the training objective, and the multi-step training in detail.

2.2.1 Training Objective

As mentioned before, we use ProphetNet(Qi et al., 2020) as our language model, which is trained with a novel self-supervised objective called future n-gram prediction. Given the training data (X, Y), where $X = \{x_i\}$, $i \in [1, M]$ is the M-length input and $Y = \{y_i\}$, $i \in [1, T]$ is the T-length output. Typically, the language model is trained to maximize the probability of the next token y_t conditioned on X and all the precedent tokens in Y. ProphetNet is different as it also predicts the future n-grams:

$$L(\theta; X) = -\alpha_0 \cdot \left(\sum_{t=1}^{T} \log p(y_t | y_{<t}, X; \theta) \right) - \sum_{j=1}^{n-1} \alpha_j \cdot \left(\sum_{t=1}^{T-j} \log p(y_{t+j} | y_{<t}, X; \theta) \right) \quad (1)$$

The first part of equation is the original language model loss while the second part is the loss from predicting the future n-grams. The parameters α and other model parameters are all consistent with open-source ProphetNet[6].

2.2.2 Multi-step Training

We propose a multi-step training procedure which allows the language model gradually acquire the generation ability of high-quality titles.

General Pre-train Pre-training is a successful technique to boost the generation quality of language models (Dong et al., 2019). ProphetNet has different open-source pre-trained versions for different languages. For example, ProphetNet-EN is pre-trained with 160GB English raw texts, including Wikipedia, news, and web texts, etc. For convenience, we use a pre-trained ProphetNet (Qi et al., 2020).

Domain-specific Pre-train The pre-trained ProphetNet has a strong ability to generate fluent text in various contexts, but we hope it can focus more on the e-commerce domain. Therefore, we collect 20GB of e-commerce data consisting of the titles and the attribute keywords for domain-specific pre-training, for instance, the title and the attribute values in Table 1. We concat the keywords as model input X, and use product title as model output Y, continuously train the model by minimizing Eq. 1 until reaching convergence.

Title Rewrite Pre-train Our task is to rewrite the seller-created titles into better quality. To

[5] https://alinlp.alibaba-inc.com/

[6] https://github.com/microsoft/ProphetNet

help reduce the gap between the pre-trained language model and our task, we continue pre-training ProphetNet with product titles. We create (pseudo title, title) pairs with tens of millions of titles we have as (X, Y), and pre-train ProphetNet by minimizing Eq. 1. As stated before, all the titles, including the bad ones, can be used as the learning target, since even the bad titles have basic knowledge about titles and still maintain a better fluency compared to the corrupted ones.

Refined Title Fine-tune In particular, the model should learn from high-quality titles about information priority. Intuitively, titles from brand owners or high-rating sellers are more reliable than the others. Online CPC, CPB performance can also be a good indicator of title quality. We combine these rules and select about 10% of all titles, which is millions, as high-quality for refined fine-tuning. We sampled 500 of them and found the portion of good titles reaches 98.0%. Similarly, we create the title pairs then train our model by minimizing Eq. 1.

We start the multi-step training from the domain-specific pre-train step and use 2 32GB Tesla V100 GPUs running for 7 days until convergence.

3 Experiment

We conduct offline and online test to evaluate the generated title in terms of accuracy, information order, fluency, and real-world profits.

3.1 Offline Accuracy

We evaluate token-level accuracy and investigate how much the multi-level shuffle module helps in model training.

Token-level Accuracy Given the golden label (original title) $\bar{Y} = \{\bar{y}_i\}, i \in [1, m]$ and the generated title $Y = \{y_i\}, i \in [1, n]$, we calculate a token-level accuracy as Eq. 2.

$$Acc = \frac{\sum_{i=0}^{\min(m,n)} \mathbb{1}(y_i == \bar{y}_i)}{\min(m, n)} \qquad (2)$$

where $\mathbb{1}$ is an indicator function which equals 1 when $y_i == \bar{y}_i$, 0 otherwise. In general, if the prediction mistake happens in the earlier steps, it will propagate the error and affect the future word prediction. Therefore, a title with a wrong beginning and necessarily wrong future tokens will obtain a very low token-level accuracy. Comparably, in PLAs, the beginning of the title is more important. Therefore, the metric somehow shows the quality of generated text as a PLAs title.

Model	**Acc_u**	**Acc_c**	**Acc_m**
Model+ unchanged	**100.0%**	38.88%	34.92%
Model+ random shuffle	92.41%	62.35%	54.61%
Model+ multi-level	93.26%	**73.32%**	**64.50%**

Table 3: Generation accuracy using different shuffle modules. **Acc** is accuracy; **u**, **c**, **m** means test data unchanged, chunk-level shuffled, and multi-level shuffled.

Baselines To examine the effectiveness of the multi-level shuffle module, we train three versions of the model using the original title as output while using different shuffled data as input: unchanged (not shuffled), random shuffle (shuffle the tokens by chance), and multi-level shuffle.

Test Data With $5,000$ selected high-quality product titles (separated from the training data beforehand), we create three versions of input data for testing: **u**nchanged, **c**hunk-level shuffled, and **m**ulti-level shuffled, and obtain the (unchanged/corrupted title, title) as the test data. We test the models and calculate the token generation accuracy on three test dataset, by which we can have a more reliable result. However, it is more convincing if a model can recover the original titles from the corrupted ones with higher accuracy.

Result From Table 3 we can observe that ProphetNet trained with multi-level shuffled data outperforms the other models on the shuffled test datasets by a large margin. The multi-level shuffle strategy achieves higher accuracy than random shuffle on all test datasets, so it does help the model generate better. Moreover, the model achieves 100% accuracy when trained and tested on the unchanged data, yet becomes the worst when tested on the corrupted titles because the model only learns making no change to the input.

3.2 Information Priority and Fluency

We examine the information order and fluency via human GSB evaluation, which means to judge the generated title as Good, Same or Bad compared to the original one. We have three PLAs marketing experts from e-commerce online marketing team. Given $1,000$ pairs of the original and generated title, every rater votes every pair with one of the GSB labels. We also provide the product image, brand, and category information to help raters resolve the core information from the title. As shown in Table 4, the generated product titles have obtained +25% and 21.9% GSB improvement compared to original titles and rule-based titles, respectively.

Baseline	**Good**	**Same**	**Bad**	**GSB**
Original	47.5%	30%	22.5%	+25%
Rule-based	37.5%	46.9%	15.6%	+21.9%

Table 4: Human evaluation. GSB=Good Rate-Bad Rate

3.3 Online Test

We run the test on Google Shopping Ads in three countries (MY, PH, SG) for two months. Before the test, we split the items into the control group and test group, then upload them to GoogleAds. In this way, the traffic of the two groups is almost even and with fairly close cost and impressions. We make sure traffic is large enough to keep stable and influential (over 100MM daily impressions). In the first month, we run campaigns and observe the gap of core metrics between the two groups. Then we update the titles into generated titles in the test group and continue running campaigns normally for another month. At last, we assess the **gap change** brought by the generated titles after online for a month.

Country	**1st CPB**	**2nd CPB**	**CPB**
MY	1.75%	-15.23%	**-16.98%**
PH	-0.19%	-12.14%	**-11.95%**
SG	-0.80%	-12.44%	**-11.65%**
Country	**1st CPC**	**2nd CPC**	**CPC**
MY	+1.81%	-1.01%	**-2.82%**
PH	+5.15%	+3.53%	**-1.62%**
SG	-0.19%	+0.17%	**+0.36%**
Country	**1st CTR**	**2nd CTR**	**CTR**
MY	-0.58%	+9.93%	**+10.51%**
PH	+0.10%	+7.01%	**+6.91%**
SG	+2.52%	+7.85%	**+5.33%**

Table 5: Online test result on Google Shopping Ads.1st means the original metric gap between control and test groups; 2nd is the gap after running generated titles. The gap change is considered as the final metric.

Our core metrics are CPC, CPB, and CTR . From Table 5 we learned that the generated titles are profitable in view of lowering the cost and bringing more conversions[7]. For example, the generated titles have brought 5.33%~10.51% CTR increment, and 11.65%~16.98% CPB drop while saving the cost of PLAs about -2%. We can see a slight CPC fluctuation in SG with a +0.36% increment, which is not hurtful given the significant positive change of CPB and CTR.

4 Discussion

Besides titles, merchants usually also have a great deal of information such as product categories, attributes, etc. We experiment further on how such information helps in model training.

4.1 Category-specific Models

To explore if a category-specific model trained only for the target category can generate better than the model trained on all categories, we train and test the model only on the “Electronics" category, one of our largest categories. In specific, **Model+EL Finetune** is ProphetNet fine-tuned only on the Electronics titles without any pre-training step, which is a basic category-specific model. **Model+Pretrain+EL Finetune** is a more advanced category-specific model, which is first pre-trained on our raw text and keywords and title pairs and titles from all categories then fine-tuned only on the Electronics titles. **Model+Pretrain+Finetune** is our standard multi-step training on all titles. As shown in Table 6, the category-specific model is weak at the generation when only fine-tuned without pretraining. However, the model trained on all categories achieves better accuracy in the target category than the model pre-trained and fine-tuned only for the target category. This may be because training on all categories can facilitate the representation learning of words shared by different categories.

Model	**Acc_u**	**Acc_c**	**Acc_m**
Model+ EL Fine-tune	77.35%	42.14%	38.34%
Model+ Pretrain+ EL Finetune	91.44%	76.33%	68.07%
Model+ Pretrain+ Finetune	**93.69%**	**77.35%**	**68.92%**

Table 6: Generation accuracy of category-specific model and proposed model.

4.2 Attribute-guided Shuffle

The most frequent circumstance in bad titles is the attribute priority. Therefore, we design an

[7] Google Shopping Ads charge by clicks, dropping CPC means saving the cost.

S1-title	olympia dz-220btx series electronic calculator
S1-gen	olympia electronic calculator dz-220btx series
S2-title	27mm aeroforce hammer with double colour handle
S2-gen	aeroforce hammer with double colour handle 27mm
S3-title	household multifunctional plastic multi-clip folding drying rack underwear socks clip drying rack baby hanger
S3-gen	multi-clip folding drying rack household multifunctional plastic underwear socks clip drying rack baby hanger
S4-title	(6 months warranty) replacement toshiba satellite t115-s1100 laptop ac power adapter charger
S4-gen	laptop ac power adapter charger replacement toshiba satellite t115-s1100 (6 months warranty)
S5-title	reflective colorful angel wings laser car stickers six-pointed star beauty body free stickers modified cool decorative
S5-gen	angel wings laser car stickers six-pointed star beauty body free reflective colorful cool decorative stickers modified
S6-title	100% authentic otterbox case symmetry series case for iphone 8 & iphone 7 (not plus)
S6-gen	otterbox case for iphone 8 & iphone 7 (not plus) symmetry series case 100% authentic
S7-title	suitable for lenovo ideapad 320s notebook charging cable 310s-14isk 15ise power adapter
S7-gen	power adapter suitable for lenovo ideapad 320s notebook charging cable 310s-14isk 15ise
S8-title	moon japanese and korean ins fresh flowers for huawei mate40pro phone case internet celebrity mate30/p40
S8-gen	phone case for huawei mate40pro mate30/p40 ins fresh flowers moon japanese and korean internet celebrity

Table 7: Samples of the original titles and the generated titles by the proposed solution.

attribute-guided shuffle strategy that creates low-quality pseudo titles by changing the positions of attributes. We expect this kind of corruption can train models to concentrate more on attribute words and then learn to arrange them better. Surprisingly, as shown in Table 8, the attribute-guided shuffle is comparative but not superior to the multi-level shuffle module, which may be because multi-level shuffle can cover various types of title issues, not only the attribute positions.

Model	**Acc_u**	**Acc_c**	**Acc_m**
Model+ attribute	91.14%	71.92%	63.17%
Model+ multi-level	**93.26%**	**73.32%**	**64.50%**

Table 8: Generation accuracy of different shuffle strategies.

4.3 Why the Optimized Titles Work in PLAs

We sample the generation titles to get a clear view of the generation quality, as displayed in Table 7. First, the core information is prioritized by putting it at the beginning of the title, especially the information that helps users quickly identify the product. For example, S1 moves the model “dz-220btx" to the behind and makes sure the product type “ electronic calculator" is visible to users. Similarly, S2, S4, S7, S8 put the product type first. The model is not moving product type to the first blindly. From S3, S6 we can see that model keeps the core attributes in front of the product type to maintain better fluency. Second, the long titles become more fluent and readable. For example, S7 generates a more natural title as a sentence. S5 and S8 properly reveal the product types earlier so that users understand immediately what is selling and make the complicated attribute list more comprehensible. Therefore, the model considers both information priority and fluency to make the product title easier to read and the visible part in PLAs more clear. It is worth mentioning that the trained model fits the titles data perfectly and only predicts the words in the original titles. Hence, the information in the generated titles is usually accurate and complete.

5 Conclusion

We present a practical solution of product title optimization for PLAs which consists of multi-level shuffling for pseudo title production and multi-step training to generate high-quality titles. It can help merchants conveniently build their own profitable title optimization systems.

References

Yeyun Gong Weizhen Qi Hang Zhang Jian Jiao Weizhu Chen Jie Fu Linjun Shou Ming Gong Pengcheng Wang Jiusheng Chen Daxin Jiang Jiancheng Lv Ruofei Zhang Winnie Wu Ming Zhou Dayiheng Liu, Yu Yan and Nan Duan. 2020. Glge: A new general language generation evaluation benchmark. *arXiv*, abs/2011.11928.

José GC de Souza, Michael Kozielski, Prashant Mathur, Ernie Chang, Marco Guerini, Matteo Negri, Marco Turchi, and Evgeny Matusov. 2018. Generating e-commerce product titles and predicting their quality. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 233–243.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2401–2410.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Huajie Shao, Jun Wang, Haohong Lin, Xuezhou Zhang, Aston Zhang, Heng Ji, and Tarek Abdelzaher. 2021. Controllable and diverse text generation in e-commerce. In *Proceedings of the Web Conference 2021*, pages 2392–2401.

Jianguo Zhang, Pengcheng Zou, Zhao Li, Yao Wan, Xiuming Pan, Yu Gong, and Philip S. Yu. 2019. Multi-modal generative adversarial network for short product title generation in mobile E-commerce. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 64–72, Minneapolis, Minnesota. Association for Computational Linguistics.