CREATER: CTR-driven Advertising Text Generation with Controlled Pre-Training and Contrastive Fine-Tuning

Penghui Wei, Xuanhua Yang, Shaoguo Liu,[*] Liang Wang and **Bo Zheng**
Alibaba Group
{wph242967,xuanhua.yxh,shaoguo.lsg,liangbo.wl,bozheng}@alibaba-inc.com

Abstract

This paper focuses on automatically generating the text of an ad, and the goal is that the generated text can capture user interest for achieving higher click-through rate (CTR). We propose CREATER,[1] a CTR-driven advertising text generation approach, to generate ad texts based on high-quality user reviews. To incorporate CTR objective, our model learns from online A/B test data with contrastive learning, which encourages the model to generate ad texts that obtain higher CTR. To alleviate the low-resource issue, we design a customized self-supervised objective reducing the gap between pre-training and fine-tuning. Experiments on industrial datasets show that CREATER significantly outperforms current approaches. It has been deployed online in a leading advertising platform and brings uplift on core online metrics.

1 Introduction

For businesses that want to promote their items and services, running online advertisements on advertising platforms is an effective way to achieve their marketing goals. With the aim of attracting users to know more about the displayed items, advertisers design ad creative (such as text, image and video). Figure 1 is an illustration that shows the creative of an ad in news feed, which contains a text and an image.

An appropriate creative design capturing user interest accurately can improve the ad's click-through rate (CTR). CTR is a key metric that quantifies the effect of an ad, because click is the precondition for any further actions such as sharing and purchase taken by users. Thus designing ad creatives that can achieve higher CTR is crucial for ad delivery.

Traditionally, advertisers need to manually design the creative of each ad, and then resort to online A/B test results to continually refine initial creative for catching user interests. Such trail-and-error

---
[*]Corresponding author.
[1]**CTR**-driv**E**n **A**dvertising **TE**xt Gene**R**ation

Figure 1: An illustration that shows the creative of an online advertisement in news feed on mobile.

process is labor-intensive and usually inefficient. In terms of the text in a creative, due to the variation characteristic of language expressions, it may need to be polished multiple times for obtaining an ideal one. To improve the efficiency of ad delivery for advertisers, especially for small advertisers that may not afford to hire professional writers, this paper focuses on automatically generating the text for an ad, and the goal is that the generated text can capture user interest for achieving higher CTR.

There are several challenges to achieve this goal. **(I)** First, it is important to choose a suitable source for generating ad texts. A straightforward source is the corresponding item's title in landing page. However, a title is usually a mixture of item attributes while may not reflect user preference. In contrast, an ad text should contain insightful and informative contents that can arouse purchasing desire of users. **(II)** Second, most of current natural language generation (NLG) models are optimized using cross-entropy criterion, which is discrepant to the CTR metric we concern. To encourage the model to generate texts achieving higher CTR, there is a great need to incorporate CTR objective into training. **(III)** Last but not least, a well-trained NLG model usually need a large amount of paired data. However it is costly to collect sufficient human-written ad texts, especially for small advertisers, thus we are faced to low-resource problem.

In this paper we propose CREATER, a CTR-driven advertising text generation approach, to address the above challenges. **(I)** First, we choose high-quality user reviews as input source for generation. Compared to titles, user reviews intuitively contain contents that reflect real experience after purchasing. We also introduce an aspect term as input control code to improve the informativeness of generated text. **(II)** Second, to explicitly incorporate CTR objective during optimizing NLG models, we make use of collected user feedback through online A/B test. Advertisers always perform online A/B test to compare two different texts of a same ad, where online CTR metric reflects the distinction between a relatively "good" text and a "bad" one. We employ contrastive learning for model optimization, which encourages our model to generate texts that can achieve high CTR. **(III)** Finally, to alleviate the low-resource problem, we make use of large-scale un-paired reviews to perform pre-training that provides warm-starting. We design a novel self-supervised objective customized to our scenario, which reduces the gap between pre-training and fine-tuning.

CREATER has been deployed online in a leading advertising platform and it achieves significant improvement on core online metrics. The main contributions of this work are summarized as follows:

• We propose CREATER for generating ad texts that capture user interest based on high-quality user reviews. We make use of online A/B test data to perform contrastive learning, which encourages the model to generate texts that achieve higher CTR.

• We propose a novel self-supervised objective to provide warm-starting with unpaired reviews, which is customized to our scenario and reduces the gap between pre-training and fine-tuning.

• Experiments on industrial datasets show that CREATER outperforms previous approaches on both automatic and human evaluation, and online results verify that it brings significant uplift on core metrics.

2   Problem Formulation

Given a source x and a control code c for an ad, where the source is a high-quality user review of the ad item, the control code is an aspect term of such review to guide generation, we aim to learn a generation model $p_\Theta(y\,|\,x,c)$ that can produce an appropriate ad text y (where Θ denotes trainable parameters of the model). Our goal is that the generated ad text can capture user interest and attract users to know more about the ad item.

3   Proposed Approach: CREATER

Figure 2 illustrates the workflow of our CREATER, and it consists of two stages. The first stage is *Controlled Pre-Training*, which learns from unpaired user reviews to provide warm-starting for low-resource scenario. The second stage is *Contrastive Fine-Tuning*, which further learns from online A/B test data that reflects user feedback, aiming to encourage the model to generate ad text that can achieve higher CTR.

3.1   Stage 1: Controlled Pre-Training

We construct a large set of user reviews as the pre-training corpus $\mathcal{D}_x$. Based on $\mathcal{D}_x$, we extract a set of aspect terms $\mathcal{D}_c$ using an off-the-shelf unsupervised model ABAE (He et al., 2017), and each aspect term is typically represented as a word.

Recall that we aim to learn a generation model $p_\Theta(y\,|\,x,c)$, while the pre-training stage only makes use of *unpaired* user reviews $\mathcal{D}_x$. To ensure that the model benefits from pre-training, we propose a novel self-supervised objective customized to our scenario, which reduces the gap between pre-training and fine-tuning. The core is that, for each review $x \in \mathcal{D}_x$, we construct an aspect-based *pseudo-target* $\tilde{y}$ from the review x and mask this segment in x. The self-supervised objective is to perform aspect-controlled generation, which aims to recover the segment $\tilde{y}$ given the masked review with the guidance of corresponding aspect term.

Aspect-Controlled Masking   For a review $x \in \mathcal{D}_x$, we tokenize it as a list of segments $[x_{\text{seg\_1}},\ x_{\text{seg\_2}},\ ...]$ based on punctuations and dependency parser, where each segment $x_{\text{seg\_i}}$ is a sub-sequence of x. Given an aspect term $c \in \mathcal{D}_c$ existed in the review x, we compute the matching score between c and each $x_{\text{seg\_i}}$ with a matching function $f(c, x_{\text{seg\_i}})$.[2] We then select the segment with highest matching score as the *pseudo-target* $\tilde{y}$ for the given pair of (source x, control code c):

$$\tilde{y} = \arg\max_{x_{\text{seg\_i}}\,\in\,x} f(c, x_{\text{seg\_i}}) \qquad (1)$$

For each triple (source x, control code c, pseudo-target $\tilde{y}$), our aspect-controlled masking strategy masks the review x by replacing its pseudo-target $\tilde{y}$ with a special word "[MASK]". Thus we transform each triple $(x, c, \tilde{y})$ to a masked one $(\tilde{x}, c, \tilde{y})$, where the masked review $\tilde{x}$ is specific to the aspect term c.
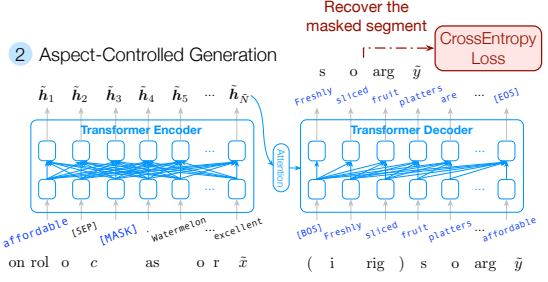
---

[2]The function $f(\cdot,\cdot)$ can either be a lexical-based one (such as similarity of sparse TF-IDF vectors) or an embedding-based one (such as similarity of averaged word embeddings).
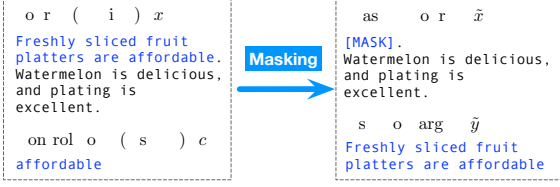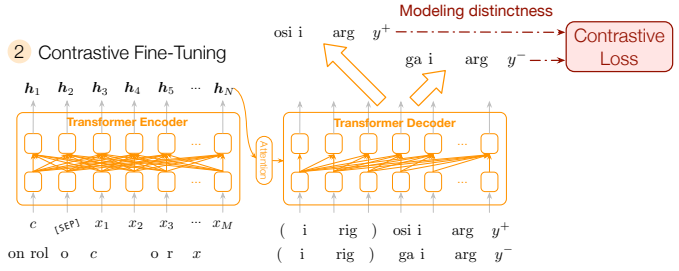
Figure 2: Overview of our proposed approach CREATER for CTR-driven advertising text generation.

Aspect-Controlled Generation Given a masked review $\tilde{x}$ with an aspect term c, our self-supervised objective is to recover the masked segment (i.e., pseudo-target $\tilde{y}$) of original review x with the controlling of c:

$$\min_{\Theta} -\log p_{\Theta}(\tilde{y} \,|\, \tilde{x}, c). \tag{2}$$

Such aspect-controlled generation enforces the model to understand the context of input masked review better. Compared to general pretraining models (Zhang et al., 2020; Lewis et al., 2020; Raffel et al., 2020), the proposed objective is customized to our scenario. The input information $\tilde{x}$ does not contain the content to be generated, improving the ability of generating abstractive contents other than simply copying from input only.

Formally, we first prepend the control code c to the masked source $\tilde{x}$, and add a special word "[SEP]" between them. We then feed the concatenated sequence $[c, \text{[SEP]}, \tilde{x}]$ into CREATER to generate the pseudo-target $\tilde{y} = [\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_{\tilde{T}}]$ (where $\tilde{T}$ denotes the length), where the model architecture is a Transformer encoder-decoder (Vaswani et al., 2017) and it is optimized via teacher-forcing:

$$\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2, ..., \tilde{\boldsymbol{h}}_{\tilde{N}} = \text{Enc}([c, \text{[SEP]}, \tilde{x}])$$

$$p(\tilde{y}_t \,|\, \tilde{y}_{0:t-1}, \tilde{x}, c) \sim \text{Dec}\left(\tilde{y}_{0:t-1}, \tilde{\boldsymbol{h}}_{1:\tilde{N}}\right)$$

$$\min_{\Theta} \sum_{t=1}^{\tilde{T}} -\log p_{\Theta}(\tilde{y}_t \,|\, \tilde{y}_{0:t-1}, \tilde{x}, c) \tag{3}$$

where $\tilde{N}$ is the length of the sequence $[c, \text{[SEP]}, \tilde{x}]$, and $\tilde{\boldsymbol{h}}_i$ is the i-th word's representation.

3.2 Stage 2: Contrastive Fine-Tuning

To incorporate CTR objective during generation, we make use of existing online A/B test data that reflects user preference. Specifically, we construct a dataset $\mathcal{D}$, where each sample is a tuple (source x, control code c, positive target y^+, negative target y^-). Both y^+ and y^- are human-written ad texts (given x and c), while y^+ achieves higher CTR than y^- during online A/B test.

Next, we start from describing a vanilla fine-tuning objective that only considers y^+. We then introduce two contrastive fine-tuning objectives which take good advantage of online A/B test data.

Vanilla Fine-Tuning A straightforward objective is to maximize the generation probability of positive target y^+:

$$\mathcal{L}_{ft} = -\log p_{\Theta}(y^+ \,|\, x, c). \tag{4}$$

Obviously, this learning objective omits the utility of negative targets.

To enhance the model's discriminative ability of ad texts with different CTR, we propose to expose the decoder to both positive and negative ad texts via modeling their distinctness. Specifically, we leverage the paradigm of contrastive learning, where the positive/negative target (i.e., ad text with higher/lower CTR) is used to construct positive/negative paired instance, and introduce two contrastive learning based objectives to fine-tune the pre-trained model.

i. Margin-based Contrastive Fine-Tuning We first propose to directly maximize the margin of generation probabilities between the positive target y^+ and the negative target y^-. This yields the following loss function:

$$\mathcal{L}_{cont} = \max\{0, -(\log p_\Theta(y^+\,|\,x, c) - \log p_\Theta(y^-\,|\,x, c)) + \gamma\} \tag{5}$$

where the margin γ is a hyperparameter. Through this loss, the optimization procedure is encouraged to maximize the probability gap of ad texts having distinct CTR.

ii. InfoNCE-based Contrastive Fine-Tuning From the perspective of representation learning, we propose a contrastive loss based on InfoNCE (Oord et al., 2018), which maximizes the similarity between source and positive target, and minimizes that between source and negative target:

$$\mathcal{L}_{cont} = -\log \frac{\exp(\text{sim}((c,x), y^+)/\tau)}{\exp(\text{sim}((c,x), y^+)/\tau) + \exp(\text{sim}((c,x), y^-)/\tau)} \tag{6}$$

where τ is temperature. $\text{sim}(\cdot, \cdot)$ is similarity function of encoder and decoder representations.

We adopt mean-pooling to the top layer of the encoder/decoder as their representations. Let $\boldsymbol{h}$, $\boldsymbol{z}^+$ and $\boldsymbol{z}^-$ denote encoder representation, decoder representations for positive and negative targets. We then add two fully-connected layers to the encoder and the decoder side respectively, transforming them to the same vector space. Thus an inner product operation is used to obtain the similarity scores:

$$\begin{aligned}\text{sim}((c,x), y^+) &= (\mathbf{W}_e\boldsymbol{h})^\top(\mathbf{W}_d\boldsymbol{z}^+)\\ \text{sim}((c,x), y^-) &= (\mathbf{W}_e\boldsymbol{h})^\top(\mathbf{W}_d\boldsymbol{z}^-)\end{aligned} \tag{7}$$

where $\mathbf{W}_e$ and $\mathbf{W}_d$ learnable parameters.

Objective The final loss of contrastive fine-tuning stage is the sum of $\mathcal{L}_{ft}$ and contrastive loss:

$$\mathcal{L}_{ft}(y^+) + \mathcal{L}_{ft}(y^-) + \alpha\mathcal{L}_{cont} \tag{8}$$

where α is a trade-off hyperparameter, and $\mathcal{L}_{cont}$ can either be margin-based or InfoNCE-based.

Comparison The advantage of margin-based loss is that it does not add extra parameters, directly incorporating CTR objective to generation probabilities. InfoNCE-based loss considers encoder representations to learn better decoder representations. Although it adds a few parameters (i.e., two fully-connected layers), they are pruned at inference. The construction of positive-negative pairs in CREATER is designed for CTR objective via user feedback, unlike recent work tackling other issues (Cao and Wang, 2021; Pan et al., 2021; Lee et al., 2021)

Dataset	Pre-training ($\mathcal{D}_x$)	Fine-tuning ($\mathcal{D}$)
# Samples	1,471,106	43,985
Avg. length of reviews	25.05	25.31
Avg. length of ad texts	N/A	13.06

Table 1: Statistics of the datasets used in our experiments. "Avg. length" means the average number of characters in a sequence (review or ad text).

4 Experiments

4.1 Experimental Setup

Datasets To our knowledge, there is no available public dataset that contains ad texts coupled with CTR information, thus we collected data on a leading advertising platform. We construct a dataset $\mathcal{D}$ where each sample is a tuple of (user review, aspect term, positive ad text$_1$, negative ad text$_2$), in Chinese, through online A/B test. Overall the user reviews are ensured to be high-quality based on rules and filtering models. Each ad text is written by human editors given the review and aspect term, covering 4,047 advertisers. More details about data preprocessing and filtering can be found in **Appendix** A.1.

We also produce a large-scale review corpus $\mathcal{D}_x$ for constructing pre-training dataset via aspect-controlled masking (§ 3.1). Table 1 lists the statistics. We split $\mathcal{D}$ with 7:1:2 to obtain the training/development/test set.

Comparative Approaches We choose two types of comparative approaches in our experiments. The first type contains *non-CTR-driven approaches*: (1) SEGEXT (Segment extraction) employs unsupervised aspect-controlled masking (§ 3.1) to return a segment of source as the ad text. If the returned segment is too short to display, we add its left or right segment based on matching score. (2) PGNET (Pointer-generator) is an RNN-based approach via copying mechanism (See et al., 2017); (3) C-PGNET improves PGNET by adding control code during decoding, which imposes on the generation gate; (4) TRM (Transformer) is the state-of-the-art architecture for text generation; (5) C-TRM improves TRM by adding control code at both encoder and decoder sides, with the help of fusion layers; (6) C-TRM-RL fine-tunes the C-TRM with reinforcement learning (RL), where an extra CTR regression model (trained on $\mathcal{D}$) is the reward estimator that produces click probability of a generated text (Hughes et al., 2019). Negative targets are used to train the reward estimator, and are not explicitly used for optimizing generation model.

The second type contains *CTR-driven approaches*. They exploit negative target y^- during training to explicitly incorporate CTR information: (1) QUALITYMODEL employs click behavior as a quality measure for paired samples (Wang et al., 2019). It first builds a CTR latent space to represent source and target, and then computes the cosine similarity between them as the quality score of the sample. Quality scores are used to weight the cross-entropy objective and reduce the probability of generating low-quality texts; (2) CONTRAMODEL is a variant of CREATER, which removes the controlled pre-training stage; (3) BART+CONTRAMODEL performs pre-training from scratch using the self-supervised objective of BART other than our proposed one, and then performs fine-tuning with CONTRAMODEL.

4.2 Implementation Details

Both the encoder and the decoder of CREATER contain four layers, and the dimension of hidden representations produced by each layer is set to 512. For fair comparison, all comparative approaches that based on Transformer employ the above architecture. For text preprocessing, we tokenize sources and targets to word sequences, and thus our CREATER generates ad texts at word-level. We restrict the max length of input as 128 words. The overall parameter size is 129M. At the pre-training stage, we employ Adafactor optimizer (Shazeer and Stern, 2018), with a mini-batch size of 4096 for training 10 epochs. Models are trained on 8 Tesla V100 32GB GPUs. We implement our approach with *PyTorch*[3] and *Transformers*[4].

In terms of the model for extracting aspect term set, during early experiments we found that the performance of CREATER is not sensitive to it and thus we employ the representative model ABAE. For matching function $f(\cdot,\cdot)$ (Equation 1) used in aspect-controlled masking for building pre-training data, we try a lexical-based (similarity of sparse TF-IDF vectors) and an embedding-based one (similarity of averaged word embeddings), and found that the performance of fine-tuned model is not sensitive to them. Thus we choose the former for simplicity.

At the fine-tuning stage, we set the mini-batch size to 1024 for 20 epochs. When the margin-based contrastive loss is used, the margin parameter γ is set to 1.0. Or if we the use InfoNCE-based contrastive

Approach	BLEU-4	RG-1	RG-2	RG-L
Non-CTR-driven Approaches				
SEGEXT	13.54	31.11	7.71	23.66
PGNET	24.85	44.79	16.76	35.21
C-PGNET	37.69	55.09	31.70	46.62
TRM	33.36	50.58	26.23	42.44
C-TRM	48.66	61.73	42.43	54.82
C-TRM-RL	50.11	62.59	42.26	55.43
CTR-driven Approaches				
QUALITYMODEL	49.89	62.67	43.85	55.84
CONTRAMODEL	51.47	63.47	43.94	56.93
BART+CONTRAMODEL	53.35	65.04	46.20	58.51
CREATER	**54.56**	**65.93**	**47.44**	**59.77**

Table 2: Main results. "RG" stands for ROUGE. Both BLEU and ROUGE scores are multiplied by 100.

loss, the temperature parameter τ is set to 1.0. We set the trade-off hyperparameter α to 1e-3 (which is searched from {1e-2, 1e-3, 1e-5}). We choose the checkpoint that has lowest perplexity on validation set as the final model. At inference time, we use beam search algorithm to generate texts, where the beam size is set to 5. The BLEU metric is evaluated using *NLTK*[5], and the ROUGE metric is evaluated using *pyrouge*[6]. All reported results of different approaches are run based on the same random seed.

4.3 Performance Comparison

Table 2 shows the comparison results, and we report BLEU-4 and ROUGE-1/2/L (positive targets are regarded as gold-standard).[7] It is natural that the approaches considering aspect terms outperform those that do not perform controlling.

CTR-driven approaches usually outperforms non-CTR-driven ones, demonstrating that exposing the model to both positive and negative targets improves generation quality. QUALITYMODEL and CONTRAMODEL represent two paradigms to incorporate CTR information. CONTRAMODEL is superior to QUALITYMODEL, which indicates that directly modeling the distinctness as an auxiliary objective is more effective than weighting the original loss.

BART+CONTRAMODEL performs better than CONTRAMODEL by adding a pre-training stage. CREATER proposes a customized controlled pre-training objective and achieves the best result. This verifies that designing a suitable self-supervised objective is crucial to improve generation.

---

[7]Our CREATER performs significantly better than the second best comparative approach at the level of $p < 0.05$.

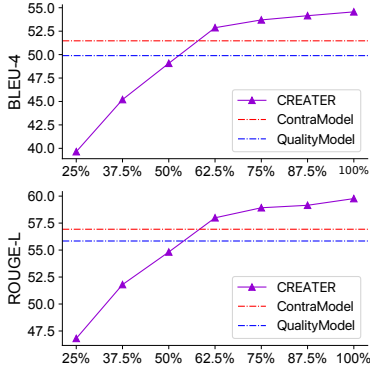| Variants of Pre-training	BLEU-4	RG-1	RG-2	RG-L
CREATER ($p(\tilde{y} \mid \tilde{x}, c)$)	**54.56**	**65.93**	**47.44**	**59.77**
w/o masking ($p(\tilde{y} \mid x, c)$)	51.24	63.65	43.74	56.94
w/o control code ($p(\tilde{y} \mid \tilde{x})$)	53.11	64.64	45.91	58.28
w/o whole pre-training	49.92	62.20	41.91	55.09

Table 3: Comparison of pre-training objectives.

Figure 3: Results with limited fine-tuning data. Dashed lines are two strongest baselines trained on whole data.

4.4 Discussion

Effect of Aspect-Controlled Masking During pre-training, aspect-controlled masking ensures the ability of generating abstractive contents other than simply copying from source. Besides, the model takes aspect terms as control codes to generated masked contents (pseudo-targets). Both the two mechanisms reduce the gap between pre-training and fine-tuning. We verify their effectiveness by removing one of two mechanisms, and the fine-tuning stage keeps unchanged. Results are shown in Table 3. The two variants are inferior to the full model, demonstrating that both of them can improve pre-training to provide better warm-starting. Aspect-controlled masking brings improvements over 3 BLEU score and 2 ROUGE score. Thus, our novel controlled pre-training objective indeed enhances the performance of advertising text generation via effective self-supervised learning on unpaired corpus.

Benefit in Low-Resource Scenario We further verify the effect of controlled pre-training when there are only limited paired data for fine-tuning. We change the size of data (from 25% to 100% of the whole training set), and compare to two strongest baselines (QUALITYMODEL and CONTRAMODEL, without pre-training) that are trained on the whole training set. As shown in Figure 3, with only half of fine-tuning data, CREATER performs on par with QUALITYMODEL, verifying the benefit of our controlled pre-training in low-resource scenario.

Variants of Contrastive Loss		BLEU-4	RG-1	RG-2	RG-L
Pre-Train	Contrastive Loss				
✓	InfoNCE-based	54.56	65.93	47.44	59.77
✓	Margin-based	54.26	65.93	47.23	59.57
✓	No	53.70	65.38	46.57	58.94
✗	InfoNCE-based	49.92	62.20	41.91	55.09
✗	Margin-based	**51.47**	**63.47**	**43.94**	**56.93**
✗	No	50.37	62.27	42.19	55.36

Table 4: Comparison of contrastive learning objectives.

Approach	Gram.	Info.	Suit.	Avg. Rank ($\downarrow$)
SEGEXT	**4.97**	2.19	1.92	4.53
C-TRM	4.95	2.69	2.44	3.65
QUALITYMODEL	4.96	2.81	2.49	3.19
CREATER	4.96	**3.21**	**3.05**	**2.09**
Human-written (high-quality)	4.99	3.60	3.22	1.48

Table 5: Human evaluation results. "Gram.", "Info.", "Suit." and "Avg. Rank" stand for grammaticality, informativeness, suitability and average rank, respectively.

Analysis of Contrastive Fine-Tuning Our CREATER exposes the model to both positive and negative targets for incorporating CTR information. Table 4 shows the comparison of two contrastive objectives. For with and without pre-training, the best-performing model is based on contrastive learning.

An interesting point is that when we perform pre-training, InfoNCE-based model achieves best performance, while margin-based model outperforms other variants if we do not pre-train the model. We suggest that InfoNCE-based loss is designed from the perspective of representation learning, and pre-training can provide better text representations compared to no pre-training. Thus in this situation the utility of InfoNCE-based model is highlighted.

4.5 Human Evaluation

An ad text will be measured from the three views: grammaticality, informativeness (whether its content reflects the key points of aspect term and the source) and suitability (whether it is suitable to be displayed). Each view is ranging from 1 to 5 (5 is the best). We randomly choose fifty samples and invite three human judgments.

Table 5 shows that CREATER performs well on most views and achieves the best ranking results among four comparative approaches, possessing the ability of generating fluent, informative and suitable ad texts. We found that the reason why the informativeness and suitability of CREATER are not as high as human-written ones is that the faithfulness of generated texts is not always ideal. We leave the improvement in future work.

14

Approach	**Source**: 水果很新鲜, 口感很好吃着非常甜, 价格优惠, 下次还会光顾 (The fruit is fresh, and it tastes delicious and sweet. The price is favorable. Will buy it next time.) **Control code**: 口感 (taste)
SEGEXT	水果很新鲜, 口感很好吃着非常甜 (The fruit is fresh, and it tastes delicious and sweet.)
C-TRM	他家的水果挺新鲜, 口感挺值的 (The fruit in this shop is really fresh, and the taste is worth the price.)
QUALITYMODEL	超喜欢他家水果, 品质好, 口感很好 (Really like the fruit in this shop, which is of good quality and tastes well.)
CREATER	份量很足, 水果很新鲜, 口感甘甜很解渴 (The fruit is a big portion and fresh. It tastes sweet, quenching your thirst.)

Table 6: Case analysis. Texts in parentheses are the corresponding contents translated to English.

Approach	CTR ($\uparrow$)	CPC ($\downarrow$)
BASE	-	-
QUALITYMODEL	+4.5%	-4.1%
CREATER	**+6.9%**	**-6.1%**

Table 7: Online results (relative improvement).

4.6 Case Analysis

We further show the generated ad texts from different approaches for case analysis. Table 6 is a case analysis that the input contains a source review with an aspect term. By comparing these generated results, We can see that the ad text generated by CREATER is more suitable to attract users. The generated phrase "sweet, quenching your thirst" is more attractive than other results like "tastes well". On the whole, the overall quality of the ad texts generated by CREATER is better than other competitive approaches.

4.7 Online Experiments

We have deployed CREATER to a leading advertising platform. Our online experiment is conducted for one-week, and all ads are displayed in mobile news feed. For the ad that containing more than one generated texts (because there may be multiple control codes), we randomly choose one of them to display. The experiment traffic covers over 12,000 advertisers, and results are computed based on over ten million impressions to ensure the confidence of online metrics.

We compare performance among the ad texts generated by CREATER, QUALITYMODEL, and those provided by advertisers (as BASE). Core metrics are CTR and cost per click (CPC): $\text{CTR} = \frac{\#\text{click}}{\#\text{impression}}$ reveals attractiveness; $\text{CPC} = \frac{\text{total cost of advertisers}}{\#\text{click}}$ reflects ad delivery efficiency. Table 7 shows that CREATER achieves significantly improvements on both CTR and CPC, verifying its effectiveness of improving delivery efficiency.

5 Related Work

Most studies focus on generating ad texts given landing page contents (Thomaidou et al., 2013). Hughes et al. (2019) employ a CTR model as reward estimator with self-critical RL, and Kamigaito et al. (2021) consider fluency, relevance and quality rewards to capture the characteristics of effective ad texts. Kanungo et al. (2021) incorporate masked language modeling with self-critical learning to improve the generation for multiple products. Wang et al. (2021) design model-based RL system that mimics real user feedback.

To model user click behavior, Wang et al. (2020) take click as a measure of text fitness and design click-based reward. Wang et al. (2019) build a CTR space to obtain sample quality that weights cross-entropy loss. Unlike these work, we directly model the distinctness of positive and negative targets, and propose a customized pre-training objective.

6 Conclusion

We propose CREATER for generating ad texts, which employs contrastive learning to encourage the model to generate texts achieving higher CTR. We design a novel self-supervised objective customized to our scenario, reducing the gap to further fine-tuning. Experiments verify that CREATER brings significant uplift on core metrics.

In future work we will take a next step to improve faithfulness, and extend the model to handle multiple aspects (Chan et al., 2021) and multiple reviews (which may be conflicting) with graph neural networks (Wei et al., 2021).

Acknowledgments

Ethical Considerations

When we apply large-scale corpora from the Web, alleviating bias issues is necessary. We make efforts from two perspectives: (1) For input reviews, we have filtering steps to remove harmful contents, and ensure that they do not have user privacy information like age and gender ("Data Collection and Filtering" of § A.1); (2) For output ad texts, we are cautious before online deployment with a risk control procedure ("Post-Processing before Deployment" of § A.1). (3) Our model does not use user privacy information like age and gender.

References

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In EMNLP.

Hou Pong Chan, Lu Wang, and Irwin King. 2021. Controllable summarization with constrained markov decision process. Transactions of the Association for Computational Linguistics.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In ACL.

J Weston Hughes, Keng-hao Chang, and Ruofei Zhang. 2019. Generating better search engine text advertisements with deep reinforcement learning. In KDD.

Hidetaka Kamigaito, Peinan Zhang, Hiroya Takamura, and Manabu Okumura. 2021. An empirical study of generating texts for search engine advertising. In NAACL.

Yashal Shakti Kanungo, Sumit Negi, and Aruna Rajan. 2021. Ad headline generation using self-critical masked language model. In NAACL.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2021. Contrastive learning with adversarial perturbations for conditional text generation. In ICLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In ACL.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In ACL.

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.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In ACL.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In ICML.

Stamatina Thomaidou, Ismini Lourentzou, Panagiotis Katsivelis-Perakis, and Michalis Vazirgiannis. 2013. Automated snippet generation for online advertising. In CIKM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. NeurIPS.

Xiting Wang, Xinwei Gu, Jie Cao, Zihua Zhao, Yulan Yan, Bhuvan Middha, and Xing Xie. 2021. Reinforcing pretrained models for generating attractive text advertisements. In KDD.

Yongzhen Wang, Heng Huang, Yuliang Yan, and Xiaozhong Liu. 2019. Quality-sensitive training! social advertisement generation by leveraging user click behavior. In WWW.

Yongzhen Wang, Jian Wang, Heng Huang, Hongsong Li, and Xiaozhong Liu. 2020. Evolutionary product description generation: A dynamic fine-tuning approach leveraging user click behavior. In SIGIR.

Penghui Wei, Jiahao Zhao, and Wenji Mao. 2021. A graph-to-sequence learning framework for summarizing opinionated texts. IEEE/ACM Transactions on Audio, Speech, and Language Processing.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In ICML.

A Appendix

A.1 More Details of Dataset Construction

Data Collection and Filtering   As mentioned in § 4.1, we construct the dataset $\mathcal{D}$ where each sample is a tuple of (user review, aspect term, positive ad text, negative ad text). The construction procedure of $\mathcal{D}$ mainly contains the following steps:

1) Collecting a set of high-quality user reviews $\mathcal{D}_x$. Firstly, a large size of reviews of e-commerce and retail items are collected. We then filter out low-quality ones via a set of rules (e.g., length constraint, repeat term constraint and harmful/abusive word vocabulary) and a spam detection model (trained based on both text contents and fraud behavior features). After this step, we obtain a review corpus $\mathcal{D}_x$ containing 1,471,106 reviews, which is also utilized to pre-training.

2) Building an aspect term set $\mathcal{D}_c$, utilized to guide generation and ensure the relevance between review contents and ad texts. According to business demands, we first construct a seed set provided by advertisers. We then expand this small set via an unsupervised extraction model ABAE (He et al., 2017), trained on the review corpus $\mathcal{D}_x$. Each aspect term is typically represented as a word. After a simple filtering

rule based on IDF to remove noise, we obtain an aspect term set $\mathcal{D}_c$ containing 991 terms.

3) Professional editors write two distinct ad texts for each given (user review, aspect term) pair. Because writing high-quality ad texts is time-consuming and labor-intensive, this procedure collects around 50,000 samples. We check the correlation between input and output via randomly sampling a fraction of all tuples written by the same editor, and remove low-quality ones. Besides, we ensure that in a paired sample the ad text does not match word-for-word to the original review.

4) Conducting online A/B test to collect user preference (i.e., CTR) on these ad texts. Traditionally, advertisers resort to this step to polish their ad texts for catching user interests. In this work we make use of these data to train contrastive learning based generation model.

5) Filtering out invalid tuples to obtain the final dataset $\mathcal{D}$. We remove outlier samples during online A/B test, e.g., the ads that do not have sufficient impressions or obtain anomalously high CTR. We also use Z-test to ensure that the CTR difference between two ad texts of same ad is significant. As a result, this dataset contains 43,985 samples and covers 4,047 advertisers.

No personal identifiable information is included in our dataset: (1) During collection, only review texts are saved, and other meta-information (such as original authors) is not collected. (2) To exclude identifying information which may be contained in texts, we employ regular expression for replacement by placeholders.

Post-Processing before Deployment Before online deployment, we have a risk control procedure to cautiously perform post-processing on the ad texts generated by models, aiming to ensure the suitability of ad texts before displaying. For instance, text contents that contain false, useless or harmful information cannot be displayed to users. Specifically, this procedure removes the texts containing non-compliant words (e.g., harmful words), and performs manual-checking on generated texts. Overall, the passing rate of generated texts is around 90% to 95%, which means that the generation models can be deployed online in industry.