Electronic Commerce Research
https://doi.org/10.1007/s10660-022-09564-6

Ad creative generation using reinforced generative adversarial network

Sümeyra Terzioğlu[1,2] · **Kevser Nur Çoğalmış**[3] · **Ahmet Bulut**[4,5]

Accepted: 8 April 2022

Abstract

Crafting the right keywords and crafting their ad creatives is an arduous task that requires the collaboration of online marketers, creative directors, data scientists, and possibly linguists. Many parts of this craft are still manual and therefore not scalable especially for large e-commerce companies that have big inventories and big search campaigns. Furthermore, the craft is inherently experimental, which means that the marketing team has to experiment with different marketing messages from subtle to strong, with different keywords from broadly relevant (to the product) to exactly/specifically relevant, with different landing pages from informative to transactional, and many other test variants. The failure to experiment quickly for finding what works results in users being dissatisfied and marketing budget being wasted. For rapid experimentation, we set out to generate ad creatives automatically. The process of generating an ad creative from a given landing page is considered as a text summarization problem and we adopted the abstractive text summarization approach. We reported the results of our empirical evaluation on generative adversarial networks and reinforcement learning methods.

Keywords Ad creative generation · Generative adversarial networks · Sequence to sequence learning

✉ Sümeyra Terzioğlu
sumeyraterzioglu@gmail.com

Kevser Nur Çoğalmış
kevser.cogalmis@izu.edu.tr

Ahmet Bulut
ahmet.bulut@acibadem.edu.tr

[1] Marmara University, 34722 İstanbul, Turkey

[2] Kuveyt Türk R&D Center, Kocaeli, Turkey

[3] İstanbul Sabahattin Zaim University, 34303 İstanbul, Turkey

[4] Carbon Health, 300 California St., San Francisco 94104, USA

[5] Acıbadem University, İstanbul 34752, Turkey

Published online: 05 May 2022

1 Introduction

A search campaign consists of a set of ad groups, each of which contains a related set of text ads and keywords as shown in Fig. 1 [1]. An advertiser places similar keywords, which are most likely to appear in user queries (or as user queries themselves), in an ad group along with text ads that contain a marketing message and a target URL. The text ads are what gets shown in ad slots. The URL is used to redirect users post click, and it is usually a product page. Text ads are ad creatives that get displayed on sponsored search results. They contain marketing messages prepared in a strict format for advertising a certain product to online users.

The success of search advertising depends on the ad creatives being relevant and attractive to users and is measured by the rate at which the user clicks on the ad creative and goes to the linked landing page. In order to meet campaign goals, various ad creatives are tested for each product. For large e-commerce companies with thousands of products, preparing these creatives and revising them according to performance needs is labor intensive. A healthcare company runs on average five campaigns for each of its landing pages. Each such campaign contains three to five ad groups. Each ad group requires three to five ad creatives since Google recommends the creation of three to five ad creatives in each ad group. Altogether, 125 creatives need to be written, which could easily overburden a small team of advertisers. For instance, the healthcare company in question needs 125 new ads written each week for its location specific campaigns. For each new location, custom marketing messages need to be created in order to capture local market dynamics. One could use generic value propositions to design ad templates. These ad templates can then be customized at runtime by dynamic keyword insertion. However, writing a creative vs. writing a creative template is essentially the same problem in theory as well as

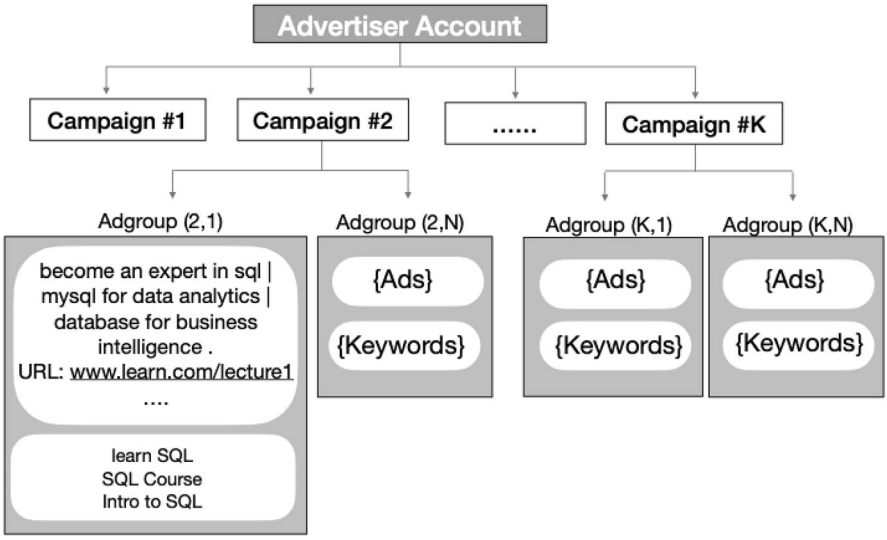

Fig. 1 A search campaign is made up of ad groups that contain ad texts and related keyword lists

in practice. Our goal is to reduce the labor by generating ad creatives automatically using the information found in landing pages.

We treated the generation of an ad creative from the landing page of a product as a text summarization task. There are existing studies and methods for text summarization. These studies fall into two main categories; extractive summarization and abstractive summarization. Extractive summarization is about generating a summary with sentences or phrases from the source text. The summary would contain sentences or phrases that represent the source text well, without using any novel words. However in abstractive summarization, novel words that are relevant to the content of the source text are used apart from the word phrases found in the source text. Because of adding novel as well as relevant phrases, it is more challenging than the extractive method. Ravi et al. established that 96% of the ad creatives contain at least one novel phrase that does not exist on the landing page based on a huge real-life corpus [2]. Therefore, we focused on abstractive summarization in this study.

Deep learning networks were promising in abstractive summarization. Generative adversarial networks (GAN) were used for novel text generation [3]. Encoder-decoder based models, i.e. the so-called sequence-to-sequence learning, and transformer-based models improved the state of the art. Bidirectional Encoder Representations from Transformers (BERT) [4], Pre-training with Extracted Gap-sentences for Abstractive Summarization (PEGASUS) [5], and Text to Text Transfer Transformer (T5) [6] are a select few of these models.

1.1 Our contribution

We proposed a mechanism to automate the generation of textual ad creatives using deep learning. Our goal is to reduce the burden of creative copy writing and to ease the selection of the best performing creatives. We combined a GAN model with reinforcement learning in order to optimize the generation of ad text. We used the information present in the landing page of a product for ad copy generation. A summary of our contributions are as follows:

1. We proposed a novel ad creative generation model for search marketing.
2. By using the product landing page as an ad creative source, we streamlined the ad creative generation process.
3. We benchmarked our model against pre-trained transformer based state-of-the-art models.

The experiments revealed that our method worked well in creating ad creatives programmatically using the descriptive information found in the landing pages. To the best of our knowledge, our study is the first to combine GANs with reinforcement learning for automating creative copy writing.

2 Related work

There are two main research areas related to our study: sponsored search advertising and text summarization using deep learning.

2.1 Sponsored search advertising

Many studies have been conducted to predict the performance of an existing ad creative [7, 8]. Zhou et al. estimated Click-Through Rate (CTR) using a deep interest network, which finally was used in the shopping site Alibaba [9]. Graepel et al. won the first place in the CTR estimation competition organized by Microsoft, and their Bayesian model became the new algorithm of the Bing search engine [10]. There are also studies on the CTR performance and optimization. Chen et al. worked on the scalability of their proposed CTR model in the recommender system of a commercial company [11].

Generating new keywords for advertising campaigns has been studied in the literature [12]. Along with generating bid phrases, effective management of the advertising budget, strategies in keyword selection, and the impact on profitability were also studied in the field of search advertising.

Du et al. conducted a study reporting the effects of keyword category choices on revenue and return on investment [13]. Vempati et al. proposed a method using deep learning for generating custom banner creatives [14]. A study conducted on Yahoo Gemini advertising platform aimed to improve the CTR performances of existing ad creatives [15]. Although there are studies on creative optimization, we did not encounter a study in the literature on how to generate ad creatives from scratch.

2.2 Text summarization

In the extractive summarization, the main task is to extract keywords and sentences from the source text, and then to generate a shorter version of the original. In the abstractive summarization, the goal is to summarize text while capturing its essence and adding novel words, which may not occur in the original text, to the final summary [16].

While most of the studies are on extractive summarization, there has been an increase in abstractive summarization due to the recent advance of neural networks. Before neural networks, some studies considered abstractive summarization with their objectives varying from capturing the essence of text content to finding representative phrases [17]. Graph-based models were also used for this purpose [18].

In general, the sequence to sequence models based on the encoder-decoder structure performed well in text summarization. Nallapati et al. performed abstractive text summarization using an encoder-decoder model with attention [19]. Additive mechanisms such as attention and pointer-generator-coverage

enhance learning from the input sequence [20]. The coverage mechanism is used to handle unwanted word repetition.

The combination of the attention mechanism and the pointer networks is effective in capturing word importance. For instance, both mechanisms were leveraged for text summarization in transformers. Transformers have a simpler architecture compared to encoder-decoder based recurrent networks, and are faster to train. They worked well in practice [21]. Due to their efficacy, pre-trained language models based on transformers were proposed. In language modeling, it was observed that models trained on large corpuses represented the language better and prevented models from over-fitting [22].

BERT and then RoBert (A Robustly Optimized BERT) [23] are effective in language modeling. BERT is a bi-directional pre-trained model on unlabeled data. It can be used for task specific training [24]. Generative Pre-trained Transformer (GPT) models that use multi-layer transformer decoder on a large unlabeled corpus achieved good results in language modeling with task specific & supervised fine tuning [25]. Transfer learning, which involves training a model on a large dataset, and then fine-tuning it for the task at hand, became more popular for NLP tasks. Another framework with a similar flavor is T5 [6], which performed well on NLP tasks. PEGASUS was utilized for abstractive summarization based on gap-sentences self pre-training phase. Their approach considered reducing or masking some sentences in the source document and generating these masked sentences from the rest of the document [5]. Rothe et al. conducted an empirical study on the performance of the aforementioned transfer learning methods [4].

A typical GAN consists of a generator and a discriminator. While the generator is used for minimizing a suitable loss function, the discriminator is used for distinguishing the input being real or fake. The back propagation from the discriminator enables the generator to produce higher quality results in order to confuse the discriminator. This approach decreases the dependence on ground truth data of the generator. And as such, it is used to address the exposure bias problem [26]. Exposure bias arises when the ground truth data is used in training, but not in testing [27].

Train and test measurement inconsistency arises when models are assessed independently in train and test phases using discrete evaluation metrics. Reinforcement Learning (RL) can be used as a solution to this problem [20]. There are RL applications solving the exposure bias problem along with the measurement inconsistency problem [28, 29].

There exist studies that combined GAN and RL [30, 31]. Wang and Lee extended a GAN with RL using reward back-propagation. They trained a generator for the production of a human-readable summary in an unsupervised and unpaired manner. The proposed model consists of a generator, a reconstructor, and a discriminator. The generator uses discriminator and reconstructor outputs as a reward to produce more accurate human-readable summaries. Their model performed well in field tests [28].

Table 1 The table of notations

Symbol	Definition
a^t	The attention distribution at timestep t.
h_i	Encoder hidden state for i^{th} token.
s_t	Decoder state at timestep t.
e_t^i	Attention distribution at timestep t for i^{th} token.
v, W_h, W_s, b_{attn}	Learnable parameters of the attention distribution.
c_t	Context vector at timestep t.
P_{vocab}	The vocabulary distribution.
V, V', b, b'	Learnable parameters of the vocabulary distribution.
P_{gen}	Generation probability to decide between copying from the source or generating a new word from the vocabulary.
σ	Sigmoid activation function.
x_t	Decoder input for timestep t.
w_c, w_s, w_x, b_{ptr}	Learnable parameters of generation probability.
$P(w)$	Final probability for given word w.
$\mathcal{L}_t$	Loss function of the generator at timestep t.
$\mathcal{L}$	Total loss function of the generator.
λ	Hyperparameter of the loss function.
$loss_t$	Final loss function at timestep t, which combines the generator and coverage losses.
N	Length of the sequence.
s_n	Discriminator score for n^{th} word in sequence.
y^s	The output of the generator for the given sequence.
K	The total number of samples in a batch.
y^{real}	Real sequence given to the discriminator.
β	Learnable parameter of the discriminator loss.

3 Proposed model

The GAN model we propose consists of a pointer-generator network with coverage [32] as its generator and a self-critic adversarial reinforce network [28] as its discriminator[1] as illustrated in Fig. 2. The details of these two key components are discussed next. In order to clarify the ensuing development, our table of notations is given in Table 1 for ease of reference.

3.1 Generator: pointer-generator network with coverage

The baseline model is a sequence-to-sequence model with attention and pointer-network with coverage [32]. Both encoder and decoder are single-layer bi-directional RNNs as shown in Fig. 3. The model combines the attention distribution and the

[1] https://github.com/yaushian/Unparalleled-Text-Summarization-using-GAN.

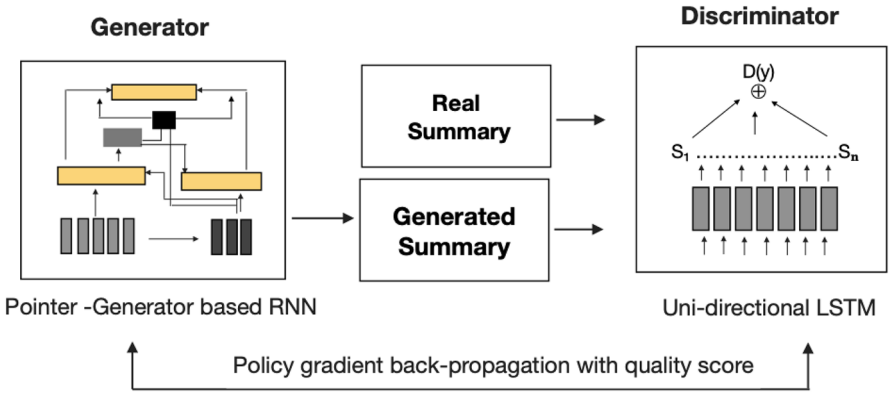

Fig. 2 A pointer-generator based bi-directional RNN as generator, and a uni-directional LSTM as discriminator

vocabulary distribution of the generated words with a soft switch called P_{gen}. It solves the inaccurate word selection problem that would commonly occur in a naive RNN-based summarization. We used this optimization in our model as well. The attention distribution is computed as:

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{attn})$$

$$a^t = softmax(e^t)$$

where a^t is the attention distribution on each step t, h_i represents encoder hidden state produced by each token y_i from source text, s_t refers to the decoder state, and v, W_h, W_s and b_{attn} are the parameters to learn. The weighted sum over the attention distribution is called the context vector c for timestep t:

$$c_t = \sum_i a_i^t h_i$$

The vocabulary distribution is computed using the decoder state s_t and the context vector c^t for timestep t as below where V, V', b and b' are the parameters to learn:

$$P_{vocab} = softmax(V'(V[s_t, c^t] + b) + b')$$

The final P_{gen} value, which is a soft switch, is computed as:

$$P_{gen} = \sigma(w_c^T c_t + w_s^T s_t + w_x^T x_t + b_{ptr})$$

where σ is the sigmoid function, x_t is the decoder input at timestep t, w_c, w_s, w_x and b_{ptr} are the parameters to learn. So with the result of P_{gen}, the final distribution to choose the next word with token word w from copying or generation is denoted as:

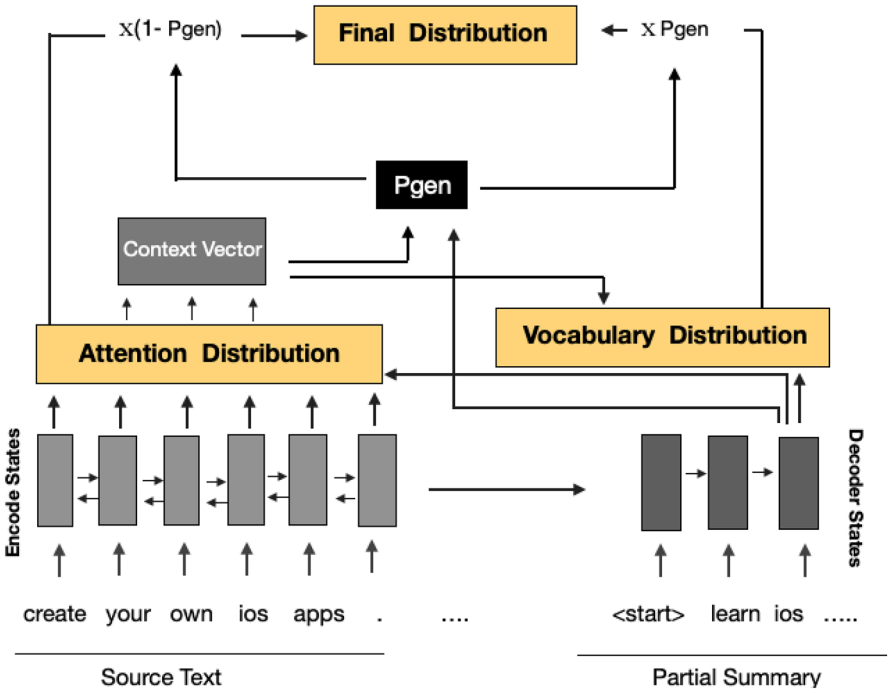

Fig. 3 Pointer-generator model. The vocabulary distribution and attention distribution are used to decide the final prediction distribution. The Pgen probability accounts for the final decision by considering the generated word from vocabulary and the copied word from the source text

$$P(w) = P_{gen}P_{vocab}(w) + (1 - P_{gen}) \sum_{i:y_i=y} a_i^t$$

The training loss of the generator is calculated as:

$$\mathcal{L}_t = -\log P(w_t^*), \quad \mathcal{L} = \frac{1}{T}\sum_{t=0}^{T} \mathcal{L}_t$$

Any unwanted word repetition, which is a by-product of the attention mechanism over emphasizing certain words, needs to be handled carefully. A coverage vector is used to avoid word repetition using a sum of attention distribution over all previous decoder time-steps. We use a coverage vector that combines the attention results with previous decoder steps and penalize the repetitive words if any:

$$c^t = \sum_{t'=0}^{t-1} a^{t'}$$

This coverage vector is fed as input to the attention distribution as follows:

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + W_c c_i^t + b_{attn})$$

Finally, the coverage loss is added to the loss function as penalty in order to avoid word repetition:

$$loss_t = -\log P(w_t^*) + \lambda \sum_i min(a_i^t, c_i^t)$$

3.2 Discriminator: self-critic adversarial REINFORCE model

One can use a generative adversarial training approach to improve the performance of the pointer-generator network. We used a self-critic adversarial reinforced model. This model has a single layer uni-directional LSTM based discriminator that takes an input word sequence y^s from the generator and that emits a score for the input sequence by considering real word sequence y^{real}. When the discriminator encounters a grammatically incorrect structure such as repetitive words coming from the generator, it penalizes the generator by assigning such sequences relatively lower scores like weight clipping as in Wasserstein GAN [33]. We use policy gradient to back-propagate the generator. The reward function is a score that indicates whether the t^{th} iteration is better or worse than the $t-1^{th}$ iteration according to the difference between their individual discriminator scores. The unidirectional LSTM takes a discrete word sequence as input and generates a score as:

$$\mathcal{D}(y^s) = \frac{1}{N} \sum_{n=1}^{N} s_n$$

where N is the length of sequence, the s_n refer to scores for each word in sequence and y^s is refers to output of Generator for given sequence. The discriminator loss is denoted by $\mathcal{D}_l$:

$$\mathcal{D}_l = \frac{1}{K} \sum_{k=1}^{K} \mathcal{D}(y^{s^{(k)}}) - \frac{1}{K} \sum_{k=1}^{K} \mathcal{D}(y^{real^{(k)}}) + \beta \frac{1}{K} \sum_{k=1}^{K} (\Delta_{y^{i(k)}} \mathcal{D}(y^{i(k)}) - 1)^2$$

where K represents the total number of examples in a batch, k denotes the k^{th} example, y^s refers to the output of generator, and y^{real} is the real sequence given to the discriminator. The last term is a gradient penalty term and y^i is the interpolated sequence of generator output. Since each score is a discrete value, we back-propagate the generator via policy gradient with the loss differences as its reward. The reward at timestep t is denoted by R_t^D and is computed as follows:

$$\mathcal{R}_t^D = \left[s_t \text{ if } t = 1, \text{ else } s_t - s_{t-1} \right]$$

In the next section, we provide the details of an empirical study we conducted in order to assess the efficacy of our model.

Table 2 A sample source document with its corresponding target document in Google's ad format

Source document	Target document (ad)
Improved awareness of your body and its needs . how to calm your mind and recover from a busy day, or prepare for the one ahead . improved clarity, focus, strength, balance, and flexibility.The history of yoga and its background . various yoga poses and routines that will benefit you the rest of your life. How to be a 'super mom' and manage the stress of daily life: yoga for busy moms . make time for you !	Yoga for busy moms | make time for you | calm your mind. Improve awareness of your body and its needs by yoga. Manage the stress of daily life.

4 Experimental evaluation

4.1 Dataset description

We used the public data of an online learning site. The landing page content for each course is used as source document. Each landing page has a title and a section named *what-you-will-learn*, which was used for constructing the target ad, i.e., the ground-truth summary, of each course. All ads were created by a domain expert and edited in order to remove redundancy, to improve its writing, and to make sure that each of them is in Google's ad format. As such, the dataset contains 4655 entries. A sample entry is shown in Table 2. In short, each target document is an ad creative in the following ad format by Google:

Headline #1 [30] \| Headline #2 [30] \| Headline #3 [30] . Description #1 [90] . Description #2 [90]

An ad in Google's format consists of three headlines, each of which contains at most 30 characters. Headlines are separated by a pipe symbol, i.e. |. The ad has two descriptions, each with a maximum of 90 characters.

4.2 Evaluation metric

There exist metrics in the literature that address grammatical correctness. However in a creative, the descriptions and the headlines are not required to be complete sentences. Therefore, we chose a metric that measures the sequence overlaps between the real summaries and the generated summaries. Specifically, we used Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score. *ROUGE* is a measure of recall since it represents the ratio of n-grams found in the original sequence that also appear in the generated sequence. Given a pair of sequences, $ROUGE_1$, $ROUGE_2$, and $ROUGE_L$ refer to their unigram, bigram, and longest common sequence overlaps respectively [34].

4.3 Training methodology

The data is randomly split into train, test, and validation sets with a split ratio of 70:15:15. In all experiments, we used a single layer bi-directional RNN with 300

Table 3 The reported *ROUGE* scores along with 95% confidence intervals correspond to F_1 scores for varying source and target sequence lengths

Model	$ROUGE_1$	$ROUGE_2$	$ROUGE_L$
Source = 100 Target =15	**41.40** [0.39,0.42]	**23.59** [0.21,0.25]	**38.84** [0.37,0.40]
Source = 300 Target =15	**43.46** [0.42,0.45]	**24.27** [0.22,0.25]	**40.53** [0.39,0.42]
Source = 350 Target = 25	**58.33** [0.56,0.59]	**833.82** [0.32,0.35]	**56.51** [0.54,0.57]
Source = 200 Target = 50	**52.67** [0.51,0.53]	**25.33** [0.24,0.27]	**50.57** [0.49,0.51]

hidden layers as a generator. The beam size parameter of beam search is set to 5. The discriminator is a single layer uni-directional LSTM with 500 hidden layers. During training, we used the Adam optimizer.

Instead of a pre-trained word embedding model, we used the whose set of words found in source and target documents as our dictionary. We used Google Colab Research and UHeM[2] computing resources in order to run Python code on GPUs.

In the source text, the average word length is 6. Each headline may have a maximum of 5 words and each description may have 15, which corresponds to a maximum of 30 characters per headline and 90 characters per description. The ad text generated by the model would have a maximum sequence length of 45- 50. The length of the source text taken from the landing page is in the 200- 250 range per sequence. This length information was taken as a reference point in setting the model parameters for input and output sequence lengths during training. In each experiment, we first trained the pointer-generator model, and then the generator-discriminator model.

4.4 Empirical results

We first run the model with the source sequence length set to 100 and the target sequence length set to 15. The scores obtained were 41.40 for $ROUGE_1$, 23.59 for $ROUGE_2$, and 38.84 for $ROUGE_L$. With increasing source length, all scores improved. When both source length and target length increased at the same time, all scores increased considerably. A higher target sequence length with a lower source sequence length performed worse than a lower target but higher source sequence length. The results are listed in Table 3; a subset of the generated summaries are shown in Table 4. Generally, the ROUGE scores tended to improve with longer sequences. Upon a close examination of our dataset, we observed that the average length of the target documents was 29. Hence, setting the target sequence length closer to this average, i.e, to 25, resulted in the highest *ROUGE* scores.

The batch size was 64 in the first three experiments of Table 3, and it was 16 in the last experiment. During the training of a GAN, the generator epoch number was set to 3000 while the discriminator epoch number was set to 1500. We gradually

[2] National Center for High Performance Computing of Turkey (UHeM).

Table 4 Generated summaries for varying source and target sequence lengths

	Example 1	Example 2
Ground-truth summary	Joomla for beginners. managing your web presence with joomla made easy	Social media marketing | marketing strategy masterclass. Learn blogging email youtube seo vlogging instagram quora pinterest podcasting. Create a youtube marketing channel and learn how to produce film and edit marketing videos.
Generated summaries for the examples above		
Source = 100 Target = 15	Joomla for beginners | web presence with joomla administrator. Use the control	Create a youtube | youtube marketing channel and learn how to produce
Source = 300 Target = 15	Joomla for managing your web presence with joomla made easy	Social media marketing | digital marketing | marketing blogging email youtube seo
Source = 350 Target = 25	Joomla for beginners. your web presence joomla joomla easy easy	Social marketing and marketing | media marketing digital marketing blogging email youtube seo instagram pinterest podcasting
Source = 200 Target = 50	Create joomla for beginners | your web with joomla made easy	Social media marketing | marketing masterclass media marketing blogging email youtube seo instagram pinterest podcasting

Table 5 *ROUGE* scores of text summarization models on the public CNN/Daily Mail dataset

Model	$ROUGE_1$	$ROUGE_2$	$ROUGE_L$
BERTShare [4]	39.83	17.69	37.01
T5 [6]	43.52	21.55	40.69
GPT [4, 25]	37.26	15.83	34.47
PEGASUS [5]	44.17	21.47	41.11
PG Network [32]	39.53	17.28	36.38

increased the discriminator epoch and observed its impact on the final results. Beyond a certain epoch, the performance started to degrade due to overfitting.

We compared the generated summaries with the ground-truth summaries in Table 4. Although different sequence lengths resulted in different *ROUGE* scores, our model generally produced relevant ad text within expected size limits. The model was also able to learn the ad format correctly without a specific directive in such regard.

The absolute *ROUGE* score beyond a certain point is not suitable as the sole determinant of the quality of the ad generated. Although the source and the target sequence lengths as 350 and 25 respectively had the highest *ROUGE* scores, the ads generated with 200 and 50 as the respective sequence lengths resulted in less repetitive creatives. We suspect that if the model accounted for the ad quality internally, the results would be more appealing to the human eye.

4.5 Comparative evaluation

We first set out to identify the best contenders for comparison using a public dataset. The *ROUGE* performances of the state of the art models on the public CNN/Daily Mail dataset[3] are shown in Table 5. The results were obtained by using pre-trained models with task specific training for summarization. The PEGASUS model outperformed all other contenders in terms of $ROUGE_1$ and $ROUGE_L$. It is a special-purpose model for summarization, and it was pre-trained with the gap sentences via self-supervised learning in the pre-training stage. Such custom training elevated its performance [5].

For comparison, we used the pre-trained transformers libraries developed by Hugging Face[4] [35]. These models are trained specifically for summarization tasks. T5 and PEGASUS models were trained on 750 GB of English-language text from the public Common Crawl web scrape. BART model was trained on the CNN/Daily Mail dataset, which contains over 300k unique news articles from CNN and the Daily Mail. After their pre-training, the subsequent training of these models on our custom dataset took a few hours. The results obtained on

[3] https://huggingface.co/datasets/cnn_dailymail

[4] https://huggingface.co/models

Table 6 The scores reported here belong to the pre-trained models from the Hugging Face transformers library after fine tuning them on our custom ads dataset. The best scores are written in bold face

Model	**ROUGE$_1$**	**ROUGE$_2$**	**ROUGE$_L$**
T5-small	44.55	30.92	37.45
BART	48.48	29.25	33.68
PEGASUS	48.39	**33.93**	41.90
Our Model Source=200 Target=50	**52.67**	25.33	**50.57**

T5, BART, PEGASUS, and our model are shown in Table 6. The generated summary samples of all models are shown in Table 7.

Among all the models considered, our model was the only model that was able to capture the meaningful use of a pipe symbol as part of the ad format as shown in Table 7. The other models were pre-trained on much larger and generic datasets; hence, they were able to produce longer summaries because of their access to a much richer domain context compared to our model, which does not have any pre-training on generic text. The domain richer context allowed these pre-trained models to capture a higher number of meaningful bigrams as shown by the higher $ROUGE_2$ scores compared to our model. However, such a creative outreach resulted in producing a higher number of novel unigrams as well. With such novel unigrams in the output, the probability of a sequence overlap with ground-truth that contains more than two grams is much lower. This is apparent in the lower $ROUGE_1$ and $ROUGE_L$ scores of the pre-trained models. With its succinct summaries, our second best model outperformed all other models in terms of $ROUGE_1$ and $ROUGE_L$ scores as shown in Table 6. It is worth mentioning that it was not possible for us to limit the size of the output of the transformers through the API provided by Hugging Face. Therefore for the sake of fairness, we compared their performance with our second best model, instead of the very best. In fact, our best model outperformed all models in all scores. From a marketer's perspective, we can state that an ad that is succinct with a few eye-catching headlines and that obeys the expected ad format is more valuable in practice compared to any other alternative that is much longer and that is devoid of such headlines. In short, the generated ad by our model spoke better to the human eye.

In order to generate an appealing ad creative, the models were expected to learn a well-defined pattern. The expected pattern was Google's ad format: an ad has three headlines, each with a maximum of 30 characters. Headlines are separated by a pipe symbol. In addition, the ad has two descriptions, each containing at most 90 characters. In order to improve model performance further in this regard, one should introduce qualitative metrics for the evaluation of the generated creatives besides such quantitative metrics as *ROUGE*. One could modify the reward function in the discriminator in a such a way that the ad format is more pronounced. This is an avenue for future work.

Table 7 Generated summaries of our model and of the pre-trained models from Hugging Face library for visual inspection

	Example 1	Example 2
Ground-truth summary	Joomla for beginners . managing your web presence with joomla made easy .	Social media marketing |marketing strategy masterclass . learn blogging email youtube seo vlogging instagram quora pinterest podcasting . create a youtube marketing channel and learn how to produce film and edit marketing videos .
Our Model Source=200 Target=50	Create joomla for beginners |your web with joomla made easy	Social media marketing |marketing masterclass media marketing blogging email youtube seo instagram pinterest podcasting
Generated summaries by pre-trained transformers libraries		
T5-small	Manage your web presence with joomla made easy. manage your web presence with joomla. manage your web presence with joomla for beginners. manage your web presence with joomla made easy	Create a youtube marketing channel. create a youtube marketing channel. create a youtube marketing channel. learn how to produce film and edit marketing videos. social media marketing masterclass. learn blogging email youtube seo vlogging instagram quora pinterest podcasting
BART	Managing your web presence with joomla made easy. learn how to create, manage, and optimize your joomala site. jooomla for beginners. managing your website with joooomsa made easy in this course you will learn how jooonsa works and how to use it to your advantage	Create a youtube marketing channel. learn how to produce film and edit marketing videos. create an instagram account. create a youtube channel. create videos on instagram. create video content on pinterest. create content on youtube. create social media marketing strategy masterclass. learn blogging email youtube seo vlogging instagram quora pinterest podcasting and more!
PEGASUS	Managing your web presence with joomla made easy. joomla for beginners. managing your web presence with joomla made easy	Create a youtube marketing channel and learn how to produce film and edit marketing videos. social media marketing : marketing strategy masterclass. learn blogging email youtube vlogging instagram quora podcasting

5 Conclusion

In this study, we proposed a new model to generate advertising creatives from product landing pages. Our proposed model consists of a pointer-generator network with coverage as its generator and a self-critic adversarial reinforce network as its discriminator. The experimental results showed that our method worked well in creating ad creatives programmatically using the descriptive information found in the landing pages. We believe that this is the first study to combine GANs with reinforcement learning in order to automate creative copy writing. In the future, we plan to advance the discriminator in order to gauge how well the generated text fits the expected ad format.

Acknowledgements The computing resources used in this work were provided by the National Center for High Performance Computing of Turkey (UHeM) under grant number 4008732020. This project was funded by Turkish National Science Foundation (Tübitak) under grant number 119E031.

References

1. Almasharawi, M., & Bulut, A. (2022). Estimating user response rate using locality sensitive hashing in search marketing. *Electronic Commerce Research, 22*, 37–51.
2. Ravi, S., Broder, A., Gabrilovich, E., Josifovski, V., Pandey, S., & Pang, B.(2010)Automatic generation of bid phrases for online advertising. In: Proceedings of the 3rd International Conference on Web Search and Web Data Mining, WSDM 2010, pp. 341–350 . https://doi.org/10.1145/1718487.1718530
3. Zhang, Y., Gan, Z., & Carin, L.(2016) Generating text via adversarial training.
4. Rothe, S., Narayan, S., & Severyn, A. (2020). Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics, 8*, 264–280.
5. Zhang, J., Zhao, Y., Saleh, M., & Liu, P.J.(2019) PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. CoRR abs/1912.08777
6. Raffel, C., Shazeer, N.M., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P.J. (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. ArXiv abs/1910.10683
7. Wang, G., Zhuo, l, Li, J., Ren, D., & Zhang. (2018). An efficient method of content-targeted online video advertising. *Journal of Visual Communication and Image Representation, 50,* 40–48. https://doi.org/10.1016/j.jvcir.2017.11.001.
8. Punjabi, S., Bhatt, P.: Robust factorization machines for user response prediction. In: International World Wide Web Conferences Steering Committee. WWW '18, pp. 669–678 (2018). https://doi.org/10.1145/3178876.3186148
9. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., & Gai, K.(2018) Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '18, pp. 1059–1068. https://doi.org/10.1145/3219819.3219823
10. Graepel, T., Candela, J.Q.n., Borchert, T., & Herbrich, R.(2010) Web-scale bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. ICML 10, pp. 13–20
11. Chen, W., Zhan, L., Ci, Y., & Lin, C.(2019) FLEN: leveraging field for scalable CTR prediction. CoRR abs/1911.04690
12. Schwaighofer, A., Candela, J.Q.n., Borchert, T., Graepel, T., & Herbrich, R. (2009) Scalable clustering and keyword suggestion for online advertisements. In: Proceedings of the 3rd International Workshop on Data Mining and Audience Intelligence for Advertising. ADKDD '09, pp. 27–36. https://doi.org/10.1145/1592748.1592753
13. Du, X., Su, M., Zhang, X. M., & Zheng, X. (2017). Bidding for multiple keywords in sponsored search advertising: keyword categories and match types. *Information Systems Resesarch, 28*(4), 711–722.
14. Vempati, S., Malayil, K.T., V, S., & R, S.(2019) Enabling hyper-personalisation: Automated ad creative generation and ranking for fashion e-commerce. CoRR abs/1908.10139

15. Mishra, S., Verma, M., Zhou, Y., Thadani, K., & Wang, W.(2020) Learning to create better ads: Generation and ranking approaches for ad creative refinement. CoRR abs/2008.07467
16. Rush, A.M., Chopra, S., & Weston, J. (2015) A neural attention model for abstractive sentence summarization. CoRR abs/1509.00685
17. Gupta, S., & Gupta, S. K. (2019). Abstractive summarization: an overview of the state of the art. *Expert Systems with Applications, 121,* 49–65. https://doi.org/10.1016/j.eswa.2018.12.011.
18. Li, W., He, L., & Zhuge, H.(2016) Abstractive news summarization based on event semantic link network. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 236–246. https://aclanthology.org/C16-1023
19. Nallapati, R., Xiang, B., & Zhou, B. (2016) Sequence-to-sequence RNNs for text summarization. CoRR abs/1602.06023
20. Keneshloo, Y., Shi, T., Ramakrishnan, N., & Reddy, C.K.(2018) Deep reinforcement learning for sequence to sequence models. CoRR abs/1805.09461
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I.(2017) Attention is all you need. CoRR abs/1706.03762
22. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X.(2020) Pre-trained models for natural language processing: A survey. CoRR abs/2003.08271
23. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V.(2019) Roberta: A robustly optimized BERT pretraining approach. CoRR abs/1907.11692
24. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K.(2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423
25. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I.: (2019) Language models are unsupervised multitask learners.
26. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y.(2014) Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. NIPS'14, pp. 2672–2680
27. Ranzato, M., Chopra, S., Auli, M., & Zaremba, W.(2016) Sequence level training with recurrent neural networks. In: 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings
28. Wang, Y., Lee, H.-Y.: Learning to encode text as human-readable summaries using generative adversarial networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4187–4195 (2018). https://doi.org/10.18653/v1/D18-1451
29. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V.(2016) Self-critical sequence training for image captioning. CoRR abs/1612.00563
30. Yu, L., Zhang, W., Wang, J., & Yu, Y.(2016)Seqgan: Sequence generative adversarial nets with policy gradient. CoRR abs/1609.05473
31. Li, J., Monroe, W., Shi, T., Ritter, A., & Jurafsky, D.(2017) Adversarial learning for neural dialogue generation. CoRR abs/1701.06547
32. See, A., Liu, P.J., & Manning, C.D.(2017) Get to the point: Summarization with pointer-generator networks. CoRR abs/1704.04368
33. Arjovsky, M., Chintala, S., & Bottou, L.(2017) Wasserstein GAN
34. Lin, C.-Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004). https://aclanthology.org/W04-1013
35. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., & Rush, A.(2020) Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6