Generating Better Search Engine Text Advertisements with Deep Reinforcement Learning

J Weston Hughes*
UC Berkeley
Berkeley, CA
jwhughes@berkeley.edu

Keng-hao Chang
Microsoft
Sunnyvale, CA
kenchan@microsoft.com

Ruofei Zhang
Microsoft
Sunnyvale, CA
bzhang@microsoft.com

ABSTRACT

Deep Reinforcement Learning has been applied in a number of fields to directly optimize non-differentiable reward functions, including in sequence to sequence settings using Self Critical Sequence Training (SCST). Previously, SCST has primarily been applied to bring conditional language models closer to the distribution of their training set, as in traditional neural machine translation and abstractive summarization. We frame the generation of search engine text ads as a sequence to sequence problem, and consider two related goals: to generate ads similar to those a human would write, and to generate ads with high click-through rates. We jointly train a model to minimize cross-entropy on an existing corpus of Landing Page/Text Ad pairs using typical sequence to sequence training techniques while also optimizing the expected click-through rate (CTR) as predicted by an existing oracle model using SCST. Through joint training we achieve a 6.7% increase in expected CTR without a meaningful drop in ROUGE score. Human experiments demonstrate that SCST training produces significantly more attractive ads without reducing grammatical quality.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**; **Reinforcement learning**; • **Information systems** → *Sponsored search advertising*.

KEYWORDS

abstractive summarization, deep reinforcement learning, seq2seq, click prediction, sponsored search, ad creative optimization

ACM Reference Format:
J Weston Hughes, Keng-hao Chang, and Ruofei Zhang. 2019. Generating Better Search Engine Text Advertisements with Deep Reinforcement Learning. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330754

*This work was completed during the first author's internship at Microsoft.

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330754

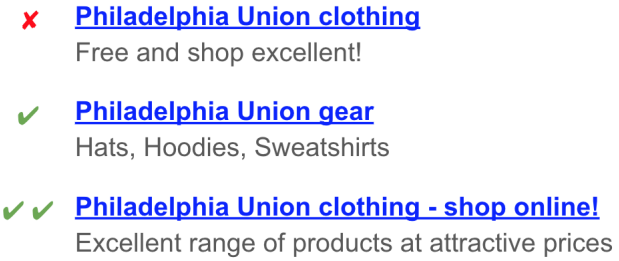

Figure 1: Above are shown three potential search engine text advertisements for a clothing website which a model might generate. The first contains nonsensical text, and is therefore unacceptable for use. The second contains a semantically and informationally correct advertisement that a human might plausibly have written. The third is both correct and attractive as an advertisement. Our approach aims to train a model to output ads which are both correct and attractive.

1 INTRODUCTION

Search engines derive a large portion of their revenue from showing text advertisements along with search results. When a user submits a query, advertisements relevant to that query (as determined by a ranking algorithm and hand- or machine-specified keywords) are displayed along with search results. If multiple advertisements are deemed relevant to a single query, advertisers may "bid" on advertisement slots in an automated auction process. Advertisers typically only pay when their ad is clicked on by a user, so it benefits both search engines and advertisers when ads are attractive and relevant [25].

Advertisements correspond to landing pages in an advertiser's domain, for example a product page on a shopping site. Many domains contain thousands of different landing pages to be advertised, while some contain as few as one. The "creative," "ad copy," or text content of these advertisements is most commonly either written by hand by the advertiser, or generated by scraping websites and automatically generating ads based on a "fill-in-the-blank" template created by the advertiser [1]. This scraping is facilitated through dynamic search advertising (DSA) platforms, which aim to automate as much of the advertising process as possible. Creating unique copy for each page can quickly become economically inefficient for larger web sites who advertise thousands of different products. Templates on the other hand are rigid and fail to adapt in the ways a

human would, and may still require the advertiser to fill in some information. In addition, smaller advertisers may not wish to employ copywriters or advertising experts to engineer their advertisements for them.

Two goals to be considered in the generation of advertisements are the correctness (both syntactic and semantic) and the clickthrough rate (CTR) of the ad. While search engine advertisements rarely include full sentences, they still must effectively convey information through natural language in a way consistent with advertising norms and user expectations (see figure 1) [1]. This correctness can be modeled as similarity to the distribution of existing human written ads conditioned on the corresponding landing pages. The clickthrough rate (CTR) or click probability of an ad is defined as

$$CTR = \frac{\text{Number of times an ad is clicked}}{\text{Number of times an ad is displayed}}$$

and measures the "attractiveness" of an ad to the users it is displayed to (CTR is more often defined over an entire campaign or other group of ads, but for our purposes it is more useful to define it on a single-ad basis). In "pay per click" payment models, this metric is directly related to profitability for both advertisers and search engines [3]. Human written ads will generally achieve syntactic correctness, but it may take a large amount of domain knowledge for a copy writer to write an advertisement with a high CTR. Template systems when used correctly also generate syntactically correct ads, but their rigidity may hurt CTR in many cases. A number of regression models have been developed to predict CTR based on features including advertisement text, query, and advertiser-chosen keywords [11, 13, 19], which can be leveraged to guide the generation process towards attractive ads.

1.1 Contributions

We describe a novel multi-input multi-output parallel encoder decoder model for jointly modeling multiple strings (described in section 3.2). We propose a novel application of Self Critical Sequence Training using a reward function not tied to the training data to maximize some desiderata (described in sections 3.5 and 3.6). We apply this method to maximize the expected clickthrough rate of search engine text advertisements (described in section 4.2). Finally, we present results based on automatic metrics and human judge ratings demonstrating the efficacy of this application (section 5).

2 RELATED WORK

2.1 Automatic Advertisement Generation

Many techniques in natural language processing have been applied to the automatic generation of advertisements. Rather than requiring an advertiser to fill in a template for each page, a product catalogue or search terms can be intelligently adapted by identifying important terms on a page or in a query and correcting capitalization, plualization, and other features which might effect syntax [1]. This solves the problem of incorrect or awkward-sounding grammar, but the process remains inflexible as all ads still have to follow the same patterns. On shopping or review sites, advertisements can be generated from existing promotional material already hosted on the site [4]. When such material is available, this is effective in creating syntactically correct ads that are geared towards selling the subject of the ad. In the search engine context, however, pre-written promotional text is not consistently available unless submitted by advertisers, defeating the purpose of generating the ad.

One solution to this problem is to generate advertisements from a product landing page [26]. This entails inferring which segments of a landing page are most relevant to or representative of the page, determining the sentiment of these segments, and generating new natural language containing the most positive segments. In [26], the relevance step is performed by finding n-grams that occur frequently together in a landing page, with the expectation that the most frequent n-grams will be the most important. The sentiment analysis step is performed by a typical Naïve Bayes classifier, and the natural language generation step uses a short list of rigid templates into which n-grams can be inserted. This approach is most similar to ours, but has major limitations. First, it's outputs still rely on a pre-built list of templates to generate all advertisements. Second, it fails to model "attractiveness" during generation - only grammatical accuracy and sentiment. To our knowledge end-to-end data-driven methods such as recurrent neural nets have not been applied to this task in the literature.

2.2 Summarization

The problem of distilling website text into a short advertisement is similar in structure to the problem of summarization. Summarization involves taking a long piece of plain text (for example a news article) as input, and outputting a much shorter text summary containing the most salient information in the input [17]. Both tasks involve inferring the meaning of the longer text, determining the most relevant information, and conveying that information in natural language. There also exist large corpora of both text-summary and landing page-advertisement pairs, from which data-driven models can be trained.

Recurrent neural network encoder-decoder models have had considerable success in summarization [15, 17, 20]. Their techniques can broadly be separated into two strategies, extractive and abstractive summarization. Extractive summarization entails copying the most salient pieces of the input to the output to create a summary [15]. This has a number of benefits: chiefly that it is easy for the model to achieve correct grammar and copy facts, and that the model learns to copy rare and out of vocabulary words which it might not otherwise be able to predict. In advertising, it is often necessary to copy important information like brand names, locations, and numbers which do not appear in the model's fixed vocabulary but are important to the advertisement.

Abstractive summarization generates new text, sometimes closely following input phrases but potentially writing novel text not appearing in the input text [20]. While requiring a more complicated

model, this allows for paraphrasing, flexible re-ordering, and potentially change of tone. The last point is particularly important for our application, as our goal is to take in landing pages not necessarily written for marketing and convert them to attractive advertisements. Extractive and abstractive can be combined through use of a copy-pointer mechanism [7, 22].

Quality of summarization is typically evaluated with the ROUGE-N and ROUGE-L scores, discrete measures of similarity between ground truth and predicted summaries [12]. The ROUGE-N score measures the overlap of N-grams between two pieces of text, while the ROUGE-L is based on the longest common substring between two pieces of text. Recently, Self Critical Sequence Training (SCST) has been used to directly optimize text evaluation metrics in the setting of summarization tasks [17]. While normal reinforcement learning is too noisy for most natural language applications, SCST applies a novel baseline to reduce variance [17, 18, 28].

3 MODEL DESIGN

3.1 Problem statement

We frame the prediction of ad titles and bodies as a sequence to sequence (seq2seq) task. We describe a landing page as a multi-word title and multi-word body

$$\mathbf{x} = x_1^T, \ldots, x_n^T, x_1^B, \ldots, x_{n'}^B$$

and a text ad as a multi-word title and multi-word body

$$\mathbf{y} = y_1^T, \ldots, y_m^T, y_1^B, \ldots, y_{m'}^B.$$

First consider the goal of semantically imitating a preexisting corpus of landing pages $\mathbf{x}^*$ and advertisements $\mathbf{y}^*$. In this task, we aim to find a joint distribution p_θ maximizing the likelihood of the examples $\mathbf{y}^*$, or minimizing:

$$L_{\text{XE}} = -\sum_{x, y \in \mathbf{x}^*, \mathbf{y}^*} \sum_{i=1}^{m+m'} \log p_\theta(y_i | y_{1:i-1}, x).$$

That is, we minimize the perplexity of the ad text conditioned on the landing page text. Ideally p_θ would capture both a quality advertising language model and also learn to extract salient information from the landing page, similar to how it performs in summarization tasks [20].

Next, consider the goal of increasing the CTR of predicted advertisements, or minimizing

$$L_{\text{CTR}} = -\sum_{x \in \mathbf{x}^*} \mathbb{E}_{\hat{\mathbf{y}} \sim p_\theta(\cdot|x)} \text{CTR}(\hat{\mathbf{y}}).$$

where $\text{CTR}(y)$ is the CTR of an advertisement y as predicted by an oracle model. This prediction is also parameterized by the query leading to the display of the advertisement, keywords determined by the advertiser, the location of the advertisement on the page, and the URL displayed with the advertisement; all of these are abstracted away in equations for simplicity, but are discussed below.

3.2 Parallel Encoder/Decoder

Our model builds on the standard recurrent neural net encoder/decoder architecture [23], which takes as input a single string of tokens and predicts a single string of tokens. Our model takes in k sequences (as in [8]), condenses them into a single latent state, and then predicts l output sequences. Here we set $k = l = 2$.

We begin by mapping each word $x_1^T, \ldots, x_n^T, x_1^B, \ldots, x_{n'}^B$ to an embedding from a matrix W_{emb}, resulting in a pair of sequences of vectors $e_1^T, \ldots, e_n^T, e_1^B, \ldots, e_{n'}^B$. These sequences are fed into two bi-directional encoder LSTMs with hidden states $h_i^{x^T} = [h_{f,i}^{x^T} || h_{b,i}^{x^T}]$, $h_i^{x^B} = [h_{f,i}^{x^B} || h_{b,i}^{x^B}]$. The last hidden states are concatenated into a latent vector s. We then predict $y_1^T, \ldots, y_m^T$ and $y_1^B, \ldots, y_{m'}^B$ using decoder LSTMs with initial state s and attending over both input hidden state sequences. The architecture is shown in Figure 2.

3.3 Attention

Our model uses Luong attention modules with intra-attention scaling [14, 17]. Each of the decoders $D \in \{y^T, y^B\}$ attends over each of the encoders $E \in \{x^T, x^B\}$. At each timestep i in each D, for each timestep j in each E, we calculate attention scores

$$e_{t,j}^E = \langle h_t^D, h_j^E \rangle$$

We then normalize these scores using past attentional scores, resulting in normalized scores

$$e_{t,j}^{E\,\prime} = \begin{cases} \exp(e_{t,j}^E) & t = 1 \\ \frac{\exp(e_{t,j}^E)}{\sum_{s=1}^{t-1} \exp(e_{s,j}^E)} & t > 1 \end{cases}.$$

Note that in the denominator above, we sum over past decoder steps, meaning that if in past steps a large part of our attention distribution was placed on a specific encoder step, in future steps this will be down-weighted, thus reducing the likelihood that the model will repeat itself [21]. Next we normalize across encoder steps

$$\alpha_{t,j}^E = \frac{e_{t,j}^{E\,\prime}}{\sum_{i=1}^{\text{len}(E)} e_{t,i}^{E\,\prime}}.$$

These α's serve both as a distribution over input words to copy from, and as the weights for calculating the context vector

$$c_{t,j}^E = \sum_{i=1}^{\text{len}(E)} \alpha_{t,i}^E h_i^E.$$

3.4 Output

At each decoder time step i, the decoder LSTM outputs a hidden state h_i^D, attention distributions $\alpha_{i,j}^{x^T}$, $\alpha_{i,j}^{x^B}$, and context vectors $c_t^{x^T}$, $c_t^{x^B}$. From this, the model predicts two distributions over potential output words, p_{vocab} and p_{copy}. The model can either predict a word using a typical softmax method, or point to a token in either of the input strings to copy as its output [22]. From $o_t^D = [h_i^D || c_t^{x^T} || c_t^{x^B}]$ we construct a distribution over our vocabulary

$$p_{\text{vocab}}(y_t^D) = p(y_t^D | u_t^D = 0) = \text{softmax}(W_{\text{out}}^D o_t^D)$$

and also a distribution over the tokens in each input string

$$p_{\text{copy from } T}(y_i^D) = p(y_i^D = x_j^{x^T} | u_t^D = 1) = \alpha_{i,j}^{x^T},$$

$$p_{\text{copy from } B}(y_i^D) = p(y_i^D = x_j^{x^B} | u_t^D = 2) = \alpha_{i,j}^{x^B},$$

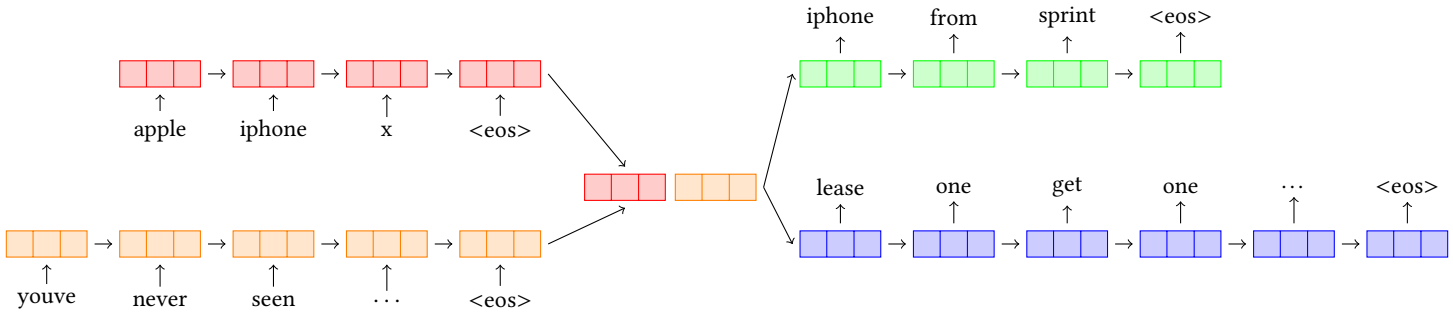

Figure 2: Our model encodes both title and body information in parallel, and then outputs first ad title and then ad body. Each decoder RNN attends over both encoder RNNs. Sttention is not pictured here.

(i.e. the attention distribution over each encoder). We also predict the three-way switch variable

$$p(u_t^D) = \text{softmax}(W_u^D o_t^D + b_u^D).$$

The final output at each step is chosen from the distribution

$$\begin{aligned} p(y_t^D) =& p(y_t^D|u_t^D = 0)p(u_t^D = 0) \\ &+ p(y_t^D|u_t^D = 1)p(u_t^D = 1) \\ &+ p(y_t^D|u_t^D = 2)p(u_t^D = 2). \end{aligned}$$

Thus the model can interleave words from a large vocabulary and words copied from the input text. We set

$$W_{\text{out}} = \tanh(W_{\text{emb}} W_p),$$

allowing the semantic relationships learned in the embedding matrix to be used in the output matrix [17]. In total, the model learns the parameters for all four LSTMs, the embedding matrices W_{emb}, W_p, and the switch parameters for each decoder b_u^D, W_u^D.

Based on preliminary human experiments (described below), we also experimented with hard masking of repeated words in each output string, such that each decoder repeats a word with probability 0. While this resulted in a slight loss in ROUGE scores, it improved readability significantly (see section 5.2).

3.5 Self Critical Sequence Training

Using the REINFORCE trick, we can compute the gradient of L_{CTR} with respect to θ [27]. The gradient

$$\nabla_\theta L_{\text{CTR}} = E_{\hat{\mathbf{y}} \sim p_\theta(\cdot|x)} \left[\text{CTR}(\hat{\mathbf{y}}) \nabla_\theta \sum \log p_\theta(\hat{y}_i | \hat{y}_{1:i-1}, \mathbf{x}) \right]$$

would normally be estimated during gradient descent as

$$\nabla_\theta L_{\text{CTR}} \approx \text{CTR}(\hat{\mathbf{y}}) \nabla_\theta \sum \log p_\theta(\hat{y}_i | \hat{y}_{1:i-1}, \mathbf{x})$$

for one $\hat{\mathbf{y}}$ sampled from the decoders. In Self Critical Sequence Training, the REINFORCE algorithm is additionally baselined by $r(\bar{y})$ for $\bar{y}$ the most likely value of p_θ, thus comparing samples from the model against its own current test-time prediction [18]. This doesn't change the expectation of the gradient, but in practice lowers the variance significantly [24]. We baseline this gradient by the arg-max of the decoders:

$$\nabla_\theta L_{\text{CTR}} \approx (\text{CTR}(\hat{\mathbf{y}}) - \text{CTR}(\bar{\mathbf{y}})) \nabla_\theta \sum \log p_\theta(\hat{y}_i | \hat{y}_{1:i-1}, \mathbf{x})$$

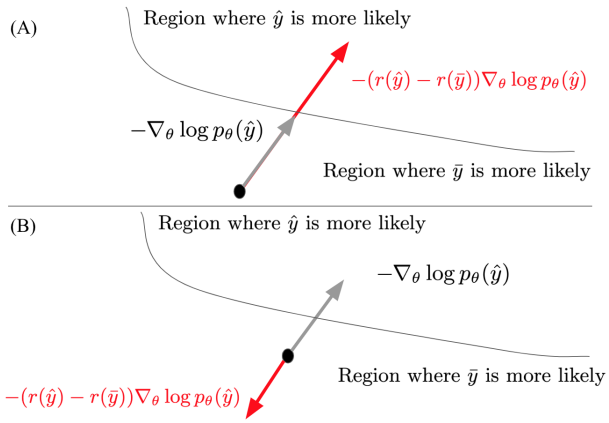

Figure 3: In Self Critical Sequence Training, the gradient towards a new sample $\hat{y}$ is scaled by the increase in reward seen if that sample is predicted instead of the model's current most likely prediction $\bar{y}$. Thus if $\hat{y}$ has a higher reward than $\bar{y}$ (as in (A)), the probability of seeing $\hat{y}$ will increase, while if the reward is lesser (as in (B)), the probability will decrease.

where $\bar{\mathbf{y}}$ is the output of the decoder arg-maxed at each step. Thus when a sample is better than the one the model would output at test time, we increase the probability of seeing that sample, and when a sample is worse, we decrease the probability of seeing it (see Figure 3).

To encapsulate both loss functions, we define the loss function

$$L_{\text{total}} = \gamma L_{XE} + (1 - \gamma) L_{CTR}$$

for some hyper-parameter $\gamma \in [0, 1]$.

3.6 Relationship to Previous Self Critical Sequence Training Approaches

Our approach is algebraically similar to those proposed by previous SCST literature ([17, 28]), but differs fundamentally because of the reward function used. In previous works, SCST has been applied to

increase the agreement between model output and a training corpus, either directly through using a measure of similarity between ground-truth and predicted output as the reward, or indirectly using a reward like a measure of salient information preserved between input and output or amount of abstractive text generated [10, 16]. Our method does the opposite: training towards sampled points with higher predicted CTR may pull the model *away* from the training corpus and cause it to predict novel advertisements with a higher click rate than those written by humans. Note that for $\gamma = 0$, our reward is completely independent of the advertisements in the training set. We choose a small value (between .02 and .1) to encourage the model to deviate from the training set while regularizing towards a language model based on the training set.

4 EXPERIMENTS

4.1 Data

Table 1: Summary Statistics of Data

Landing Page Title Avg. Length	8.18
Landing Page Body Avg. Length	809.33
Advertisement Title Avg. Length	6.70
Advertisement Body Avg. Length	10.80
# Training Examples	303,387
# Validation Examples	10,000
# Test Examples	10,000

We trained our models using Bing Ads data. From a sample of 600,000 landing page-ad pairs, we removed examples such that each domain appeared at most 5 times. This step was important as a small number of advertisers comprise a large fraction of ads, many of which follow the same templates which the model could be forced to copy. This resulted in a dataset of 323,387 examples, which we separated into training, validation, and test sets of sizes 303,387, 10,000, and 10,000. No domain appeared in multiple sets. Landing pages were normalized to contain no punctuation and be all lower case. Ads were similarly normalized, keeping key punctuation including "?" and "-" which appear often in advertisements and have important semantic meaning.

We used a dictionary of the 30,000 most common words appearing across landing pages and advertisements. During tokenization, all words not appearing in the dictionary were assigned to an OOV token. OOV's in the landing page were indexed to facilitate creation of the copying distributions described above [22].

4.2 Training & Reward

We pre-trained our model to directly optimize L_{XE} using the ADAM optimizer with default parameters [9]. The best-performing weights of this model were then used to initialize a model trained to minimize L_{total} on the training set using Self Critical Sequence Training. We trained models with different γ values, and in some cases used an early stopping procedure described below. For the SCST training phase, we used ADAM with a learning rate of 10^{-4}, following [17].

To estimate CTR during training, we used an oracle model pre-trained to predict CTR on text ads. This oracle was treated as a black box, and its weights were neither re-trained nor observed during SCST training. Our model simply queried the oracle by submitting an advertisement and metadata, and the oracle predicted the CTR of the advertisement. As the oracle, we used a large-scale logistic regression model from the Bing Ads ranking pipeline, similar to and using the same training set as the one described in [13]. The training set consisted of millions of query-advertisement pairs, as well as the user click corresponding to each pair. Features included millions of cross-feature indicator variables, binary variables activated when a pair of words appear in the query and advertisement (for example, "car" in the query and "vehicle" in the advertisement). For simplicity, we set the user query equal to a keyword corresponding to each landing page submitted by the advertiser, and assumed that all advertisements appear in the same location (as the first "search result," or "mainline-1" in advertising terminology).

We found that training with low values of γ resulted in a pattern where expected click rate would increase steadily with roughly constant ROUGE score, until some threshold where the CTR reward would "win out" over the cross-entropy loss, at which point the reward grew rapidly while the ROUGE dropped. Examination of model outputs showed that this resulted from severe over-fitting to the oracle model, generating nonsense containing repeated "attractive" words. Because such ads don't appear in the oracle's training set, the oracle's predictions are unreliable for these ads. To mitigate this effect we added early stopping based on ROUGE-L score. Whenever the validation ROUGE-L of either title or body drops by more than .8%, training ends and the last checkpoint is used (.8% was chosen to be large enough to allow for normal variation in the training process).

5 RESULTS

Using SCST to train our model significantly raised the accuracy and attractiveness of the generated advertisements, as measured both by automatic methods and through a number of human experiments designed to demonstrate that we are not over fitting to the oracle model. We evaluated the training procedure for $\gamma \in \{.1, .05, .02\}$, training all models from the same pre-trained checkpoints.

5.1 Automatic evaluation

Results of automatic evaluation are discussed in Table 2. We report expected click rates predicted by the oracle model as well as a separate neural net model trained on the same database from a different time period [13]. We also report ROUGE-L scores between the ground truth advertisements and those produced by the model. All statistics are averaged over a 10,000 example test set not viewed at any point during development.

We found $\gamma = .05$ to be the optimal value. Setting $\gamma = .02$ caused the ROUGE-L to decrease too quickly, overfitting to the oracle and generating examples with worse grammar, and thus hitting the early stopping cutoff too quickly. Conversely, setting $\gamma = .10$ resulted in a smaller increase in click rate. We found that $\gamma = .05$ provideed the best performance at convergence in terms of expected

Table 2: Predicted click rates of human written "ground truth" advertisements and those produced by various models, and ROUGE scores between ground truth and generated ads. Metrics are calculated over entire test set.

Model	Click Rate (Oracle Prediction)	Click Rate (DNN Prediction)	Title ROUGE-L	Body ROUGE-L
Human Written Ads	13.19%	14.56%	100	100
Baseline Model	10.81%	12.57%	35.4	22.9
SCST ($\gamma = .02$)	11.42%	13.14%	36.2	22.3
SCST ($\gamma = .05$)	**11.54%**	**13.17%**	**36.3**	22.3
SCST ($\gamma = .10$)	10.95%	12.77%	35.6	**23.0**
Baseline Model (repeats masked)	10.29%	12.28%	34.2	22.7
SCST ($\gamma = .05$, repeats masked)	10.73%	12.73%	34.8	21.9

click rate, while not meaningfully decreasing ROUGE scores and cross-entropy.

The oracle predicts an increase of up to 6.7% in click rate when using SCST compared to baseline training when no masking is employed, without meaningful difference in ROUGE score. The DNN model predicts a 4.8% increase in click rate, suggesting that the method is not just overfitting to the oracle model but rather genuinely pushing the model to generate more attractive ads. The ROUGE-L scores do not differ significantly between the trained models, suggesting that the grammatical accuracy is consistent across models. The fact that ROUGE scores don't significantly change while click rates increase suggests that the model is optimizing within the space of advertisements written by humans, towards those with higher click rates.

After training the first set of models we noted that many of the generated ads contained repeated words and phrases despite the use of intra-temporal attention (this is a common problem in sequence to sequence settings [2]). This was more common after SCST training, probably because the oracle model was partially "tricked" by repeated attractive words in out of domain examples. To combat this, we additionally implemented a hard constraint on the model, disallowing any repeated words in the title or body, as suggested in [2]. Expected click rates are reduced when using this constraint, but follow the same upward trend when SCST is applied.

5.2 Human evaluation

We conducted two sets of human evaluation trials to evaluate the quality of the generated advertisements. The results are reported in Tables 4 and 5. In both trials, we recruited trained human judges to rate and compare our generated advertisements. In the first set of experiments, judges were shown a single advertisement and the corresponding landing page, and were asked the following factual and subjective questions:

- In the above ad, are there any unnecessary repeating words or phrases? [Yes/No]
- Does the above Ad Title and Ad Copy seem like it was written by a human? [Yes/No]
- Please a provide a rating about the Ad Title and Ad Copy [Good (Perfect grammar) / Fair (Slight grammar errors or repeats) / Bad (Major grammar errors or repeats) / Nonsense or Broken (Including foreign language or landing page not loading)]

The same 2000 data points were used to evaluate each model, the examples were shuffled together so judges could not determine which examples came from each model, and some ads were repeated to ensure factual questions were answered consistently.

Disallowing repeated words improved ratings both on grammar and likeness to human advertisements. Between the two masked models we evaluated, the SCST-model-generated ads were more often deemed human-like (83.0% vs 78.8%, $P = .000728$), showing that we are properly regularizing towards the distribution of human-written ads. The portion of ads with "Good" grammar fell slightly but not significantly. Based on these results we conclude that SCST does not reduce the grammatical quality of generated ads, and thus doesn't hurt the semantic language model learned during pre-training. (The relatively high amount of "Nonsense/Broken" ratings across all ads is mainly due to expired landing pages and foreign language examples in the test set. Human-written and model-generated ads were deemed broken at similar rates.)

In the second set of experiments, advertisements for 1000 landing pages from the two repeat-masked models were compared side-by-side for quality. Results are shown in Table 5. The two ads were displayed along with the keyword/query associated with the landing page. Judges were asked

- Which Ad Copy do you like more given the Query? [Ad Copy #1 / Ad Copy #2 / Ads are identical]

Ads were shown in random order. We found that judges preferred the SCST generated ads significantly more often than the baseline model's ads ($P = 1.655 \times 10^{-5}$). Considering the results of both sets of experiments, SCST increases how often an ad is preferred while maintaining the same grammatical accuracy as the initial model.

Landing Page 1		
painting contractors gonzalez painting and waterproofing inc hialeah fl painting contractors gonzalez painting and waterproofing inc hialeah fl contact gonzalez painting and waterproofing inc for painting services in miami gonzalez painting and waterproofing inc delivering quality painting services 999 999 9999 call now for office painting services …		
True Ad	**Baseline**	**SCST**
House painting SVCs - trust a qualified professional Get professional reliable service on time every time call us first	**Miami painting** Get your home done now call us for a quote on services	**Miami painting contractor - jobs done right the first time** Get your home done now fair prices honest work call us
Landing Page 2		
lawn care service k g lawn care louisville lawn care service k g lawn care louisville ky k g lawn care provides lawn care services in louisville we put the care in lawn care 999 999 9999 call us for free estimates quality lawn care services in louisville visit website for lawn care maintenance services call …		
True Ad	**Baseline**	**SCST**
Grass cutting services - Take pride in your landscape Professional grass cutting service call now for a quote on services	**Louisville lawn care - Honest quotes and quality work** Keep your trees healthy happy quality service call today	**Louisville lawn care - Keep your landscape smooth** Get your lawn back with our expert staff call us today
Landing Page 3		
mountain view corvette restoration repair and maintenance mountain view corvette restoration repair and maintenance corvettes by north star automotive is committed to providing our clients with exceptional auto service for corvettes about us services contact us menu about us services contact us we put corvettes in the winners …		
True Ad	**Baseline**	**SCST**
Auto engine repair - Over 30 years of experience Car engine repair services at great prices call us in Mountain View	**Corvette repair - Serving mountain view** Quality Corvette repair services call our staff in Mountain View	**Corvette Repair** Call our trusted corvette dealer for a free shuttle service today
Landing Page 4		
trampolines in centennial colorado the playground guys trampolines in centennial colorado the playground guys the playground guys offers trampolines by springfree located in centennial colorado centennial co 999 999 9999 products trampolines basketball systems locations promos special offers floor model clearance about about us testimonials locations trampolines basketball systems special offers floor model …		
True Ad	**Baseline**	**SCST**
Trampoline Aurora - Serving Denver Centennial World's safest trampoline variety of shapes sizes to fit any yard call now	**Trampolines - Call us today** Trampolines for all ages call us today to schedule an appointment	**Trampolines in Centennial - Call us today** Trampolines for all ages sizes in Centennial call us today
Landing Page 5		
used peugeot partner for sale rac cars used peugeot partner for sale rac cars we have 59 used peugeot partner cars for sale throughout the uk from rac cars approved dealer click here to find a great deal login register used cars search by location search rac approved dealers search dealers…		
True Ad	**Baseline**	**SCST**
Used partner - Find the right car - Call now Search for quality used Peugeot partners search buy sell with RAC cars	**RAC cars for sale - find the best Peugeot partner** Find the best Peugeot partner for sale in stock now	**Used Peugeot partner - RAC cars for sale** RAC cars for sale throughout the UK from 59 month get a free quote today

Table 3: Comparison of advertisements for the same landing page. For each page, we show a prefix of the scraped landing page text, the human-written ad from the training corpus, the ad produced by our baseline model, and the ad produced by the SCST model.

Table 4: Results of human assessments of the same models on 2000 data-points. Human judges were asked to assess whether advertisements contained grammatical errors, report their confidence that a human wrote the advertisement, and report the overall quality of the advertisement.

Model	Unnecessary repeating words or phrases	Seem like it was written by a human	Rating of Ad Title and Body grammar			
			Good	Fair	Bad	Nonsense/Broken
Human Written Ads	8.7%	88.6%	71.6%	13.9%	1.7 %	12.6%
Baseline Model	17.6%	74.7%	64.7%	20.8%	2.8%	11.4%
SCST (γ = .02)	23.5%	74.3%	59.5%	22.9%	4.5%	13.0%
SCST (γ = .05)	25.2%	73.8%	59.7%	22.8%	4.5%	13.0%
Baseline Model (repeats masked)	3.8%	78.8%	70.5%	15.7%	1.0%	12.3%
SCST (γ = .05, repeats masked)	7.8%	83.0%	68.7%	15.9%	2.2%	13.0%

Table 5: Pairwise comparison between the baseline and SCST model. Human reviewers were asked to choose between two different advertisements side-by-side corresponding to 1000 queries and URLs.

Model	Percentage that prefer (95% CI)
Baseline	38.5 (38.2-38.9)
SCST	51.5 (51.2-51.9)
Ads are identical	10.0 (9.7-10.4)

5.3 Generated Examples

In Table 3 we include representative examples of advertisements generated by baseline and SCST models. Applying SCST causes the model to include more promises of free services, calls to action, and numbers. All of these are known to increase the attractiveness of advertisements [25].

6 DISCUSSION

The major contribution of this work is the novel application of Self-Critical Sequence Training to maximize a non-differentiable reward unrelated to the match between the training and predicted data points. Our method allows the balancing of two important concerns through varying the γ parameter: how "realistic" the output is (measured using a cross-entropy loss) and how well the output maximizes its non-differentiable reward. In doing so, one can control how far the model strays from the training data distribution in pursuit of the reward function. Applying early-stopping allows for application of harder constraints on deviation from the distribution.

Particularly interesting is the fact that using SCST generates ads deemed more human-like than using a model trained directly on a corpus of human-written ads. With the correct hyper-parameters, ROUGE scores are not decreased. In other words, the ads are drawn from an equally or more human-like distribution. They also achieve a higher predicted click rate, suggesting that the model is optimizing click rate within the distribution of human-like ads it learns from the training set, finding a part of the distribution with the highest expected click rate.

The most significant drawback of this method is its inability to obey the important but often complicated hard constraints required in advertising and most other real-world applications. One clear example of this is the model's ability to generate a false advertisement. While the cross-entropy loss should encourage the model to generate ads consistent with the corresponding landing pages, the reward may push it towards generating phrases like "free shipping" or "30% off all purchases" because these phrases are attractive, regardless of whether they are true or not. One partial solution is to run beam search and generate multiple ads, which can be rated for accuracy.

As we work to bring this method into production in the Bing Dynamic Search Ad pipeline, we are building an ensemble model using both SCST and baseline models and more traditional template methods to generate a set of ads to select from. These are rated both on expected click rate and attractiveness, as well as using hard decision rules to eliminate false claims and other potential problems. Thus we can take advantage of gains from the SCST model when it does obey hard constraints, and fall back to more rigid systems when SCST fails. Initial results suggest that much of the attractiveness added will not be filtered out by this system.

Another challenge this method faces comes from the oracle function. If the employed oracle function gives high scores for points outside of its training domain, SCST may erroneously push towards samples which don't look like real data but which it deems to be

attractive, for instance the attractive word "free" being repeated several times out of context. We solved this through balancing γ and applying early stopping and hard constraints on output, but a carefully designed oracle function might more effectively solve this problem. This might be achieved by quantifying the uncertainty of the oracle in terms of the input, or by designing an oracle function which fully encapsulates what "normal" data looks like. Jointly training a discriminator in a GAN-like framework would further encourage the model to produce realistic examples, but a strong discriminator would likely push advertisements too be as attractive as human ads, but not moreso.

Using a human-in-the-loop or online process as the reward would cleanly solve this problem, but brings a number of practical challenges. If this process is found to have a significant effect on advertising revenue, a model could be trained with live data, continuously serving newly generated ads to customers and using their clicks as a reward signal. This requires a high degree of confidence that the model won't generate low quality or incorrect ads, and would also require a large amount of infrastructure development. Still, this would represent a gold-standard signal to be optimized, and perfectly model the tradeoff between attractivity and correct syntax.

While we applied this method to advertising, its relevance is far-reaching. Similar design problems include the synthesis of novel proteins and molecular drugs satisfying specific biological constraints and objectives [6], and image style transfer [5]. While the methods currently employed in these areas are different from the ones presented here, they face the same challenges of trading off hard and soft constraints, and over-fitting to oracle functions after leaving their training domain.

7 CONCLUSION

We applied Self Critical Sequence Training to the problem of generating attractive search engine advertisements, using a novel method through which we optimized a non-differentiable metric not related to similarity between output and training data. We demonstrate that this method improves the expected click rate and human-rated attractiveness of advertisements generated by a sequence to sequence model. Furthermore, this gain comes without any significant loss in syntactic or semantic accuracy. Future work could include using this method in conjunction with a generative model to push the output generation towards desirable features, or to simply train more diverse oracle models to add further constraints to the output.

8 ACKNOWLEDGEMENTS

We're thankful to Yajuan Duan, Hao Wang, Fei Sun, Yadi Liu and Jiusheng Chen for providing guidance on using Bing Ads click prediction models, and Chris Song and Amy Liu for their guidance on parsing text ads landing pages. Additionally, we're extremely grateful to Jim Cao and Rajesh Koduru for their effort in productionizing this work, and to Tracy Ortman for her contribution to setting up human evaluation studies. Finally, a huge thanks to David Ku for introducing the first author to the team.

REFERENCES

[1] K. Bartz, C. Barr, and A. Aijaz. Natural language generation for sponsored-search advertisements. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, EC '08, pages 1–9, New York, NY, USA, 2008. ACM.
[2] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell. Language models for image captioning: The quirks and what works. *CoRR*, abs/1505.01809, 2015.
[3] D. C. Fain and J. O. Pedersen. Sponsored search: A brief history. *Bulletin of the american Society for Information Science and technology*, 32(2):12–13, 2006.
[4] A. Fujita, K. Ikushima, S. Sato, R. Kamite, K. Ishiyama, and O. Tamachi. Automatic generation of listing ads by reusing promotional texts. In *Proceedings of the 12th International Conference on Electronic Commerce: Roadmap for the Future of Electronic Business*, ICEC '10, pages 179–188, New York, NY, USA, 2010. ACM.
[5] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.
[6] R. Gómez-Bombarelli, D. K. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *CoRR*, abs/1610.02415, 2016.
[7] Ç. Gülçehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio. Pointing the unknown words. *CoRR*, abs/1603.08148, 2016.
[8] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 241–248, New York, NY, USA, 2016. ACM.
[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
[10] W. Kryściński, R. Paulus, C. Xiong, and R. Socher. Improving abstraction in text summarization. *arXiv preprint arXiv:1808.07913*, 2018.
[11] R. Kumar, S. M. Naik, V. D. Naik, S. Shiralli, S. V.G, and M. Husain. Predicting clicks: Ctr estimation of advertisements using logistic regression classifier. In *2015 IEEE International Advance Computing Conference (IACC)*, pages 1134–1138, June 2015.
[12] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. *WAS*, 2004.
[13] X. Ling, W. Deng, C. Gu, H. Zhou, C. Li, and F. Sun. Model ensemble for click prediction in bing search ads. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 689–698. International World Wide Web Conferences Steering Committee, 2017.
[14] M. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.
[15] R. Nallapati, F. Zhai, and B. Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *CoRR*, abs/1611.04230, 2016.
[16] R. Pasunuru and M. Bansal. Multi-reward reinforced summarization with saliency and entailment. *CoRR*, abs/1804.06451, 2018.
[17] R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.
[18] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. *CoRR*, abs/1612.00563, 2016.
[19] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: Estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 521–530, New York, NY, USA, 2007. ACM.
[20] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. *CoRR*, abs/1509.00685, 2015.
[21] B. Sankaran, H. Mi, Y. Al-Onaizan, and A. Ittycheriah. Temporal attention model for neural machine translation. *CoRR*, abs/1608.02927, 2016.
[22] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368, 2017.
[23] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.
[24] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. 2018.
[25] S. Thomaidou. *Automated Creation and Optimization of Online Advertising Campaigns*. PhD thesis, Ph. D. thesis, Department of Informatics, Athens University of Economics and Business, 2014.
[26] S. Thomaidou, I. Lourentzou, P. Katsivelis-Perakis, and M. Vazirgiannis. Automated snippet generation for online advertising. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, CIKM '13, pages 1841–1844, New York, NY, USA, 2013. ACM.
[27] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.
[28] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.