Learning to Create Better Ads: Generation and Ranking Approaches for Ad Creative Refinement

Shaunak Mishra
Yahoo Research
shaunakm@verizonmedia.com

Manisha Verma
Yahoo Research
manishav@verizonmedia.com

Yichao Zhou
University of California, Los Angeles
yz@cs.ucla.edu

Kapil Thadani
Yahoo Research
thadani@verizonmedia.com

Wei Wang
University of California, Los Angeles
weiwang@cs.ucla.edu

ABSTRACT

In the online advertising industry, the process of designing an ad creative (*i.e.*, ad text and image) requires manual labor. Typically, each advertiser launches multiple creatives via online A/B tests to infer effective creatives for the target audience, that are then refined further in an iterative fashion. Due to the manual nature of this process, it is time-consuming to learn, refine, and deploy the modified creatives. Since major ad platforms typically run A/B tests for multiple advertisers in parallel, we explore the possibility of collaboratively learning ad creative refinement via A/B tests of multiple advertisers. In particular, given an input ad creative, we study approaches to refine the given ad text and image by: (i) generating new ad text, (ii) recommending keyphrases for new ad text, and (iii) recommending image tags (objects in image) to select new ad image. Based on A/B tests conducted by multiple advertisers, we form pairwise examples of inferior and superior ad creatives, and use such pairs to train models for the above tasks. For generating new ad text, we demonstrate the efficacy of an encoder-decoder architecture with copy mechanism, which allows some words from the (inferior) input text to be copied to the output while incorporating new words associated with higher click-through-rate. For the keyphrase and image tag recommendation task, we demonstrate the efficacy of a deep relevance matching model, as well as the relative robustness of ranking approaches compared to ad text generation in cold-start scenarios with unseen advertisers. We also share broadly applicable insights from our experiments using data from the Yahoo Gemini ad platform.

CCS CONCEPTS

• **Information systems → Online advertising**.

KEYWORDS

Online advertising; A/B testing; sequence2sequence; ad creatives.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10.
https://doi.org/10.1145/3340531.3412720

ACM Reference Format:
Shaunak Mishra, Manisha Verma, Yichao Zhou, Kapil Thadani, and Wei Wang. 2020. Learning to Create Better Ads: Generation and Ranking Approaches for Ad Creative Refinement. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3340531.3412720

1 INTRODUCTION

The image and text used for an online ad (collectively called an ad creative) can be influential in targeting online users on a large scale. Large businesses (advertisers) typically employ creative strategists to design ad creatives; these creative strategists may conduct market research to see trending themes and also gather insights from past ad campaigns in related product categories. Such advertiser specific creative customization is mostly a manual, expensive, and time consuming process. In contrast, small businesses typically resort to free online tools, *e.g.*, stock image libraries [2], and generic creative insights [3] to compile ad images and text; such tools can reduce the time to design creatives but tend to be generic (*e.g.*, lacking in business-specific customization). Once the ad creatives are ready, both large and small advertisers need to conduct online A/B tests to validate the effectiveness of their creatives, and subsequently discard low performing creatives from their ad campaigns. In addition, to reduce the chances of online users getting tired of seeing the same ad repeatedly on a particular website (*i.e.*, ad fatigue [23]), advertisers need to frequently go through the design→A/B test→refresh ad creatives cycle . Again, such cycles tend to be time consuming and there is an emerging need for data-driven approaches to speed up the whole process of designing and refreshing creatives.

In this paper, we highlight a key observation that accelerates the above creative design process, and can be explained as follows. Advertisers typically test their creatives via A/B tests in ad platforms (*e.g.*, Yahoo Gemini, Facebook Ads), *i.e.*, they try out a set of creatives on online users in a controlled setup such that the click-through-rate (CTR) performance [5] difference across the creatives can be solely attributed to the ad text and image. However, advertisers conduct and learn from such A/B tests in isolation as illustrated in Figure 1. As shown, advertiser 1 who is an internet service provider, may learn via an A/B test that having human elements in the ad image works better than having gadgets in the image (since the ad text is same across the two creatives in the example, the performance difference can be attributed to the ad images). Via a separate A/B test, a different advertiser N (selling

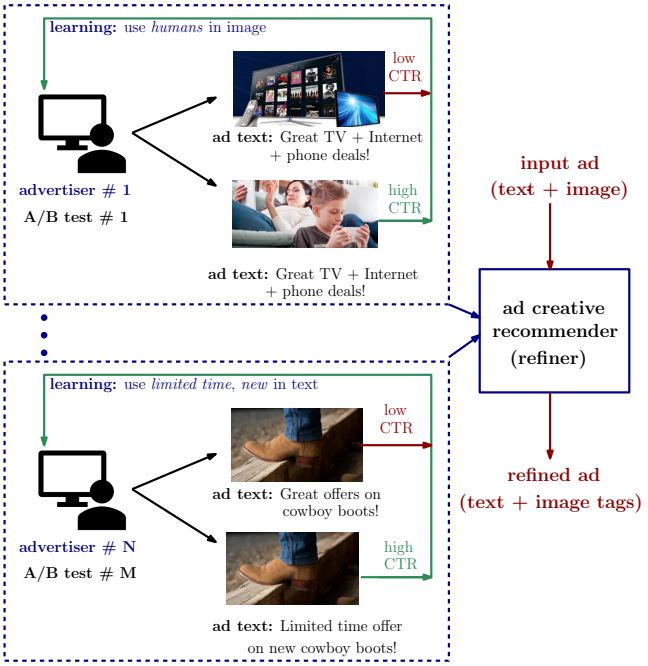

Figure 1: Ad creative refiner based on parallel A/B tests done by multiple advertisers. Advertiser 1 may learn in isolation that having human elements leads to better CTR than multimedia images; while advertiser N may learn that using "limited time" in ad text works better than "great". The proposed refiner collects data across A/B tests to recommend ad text and image refinements for a given input ad creative.

boots) may learn that using *new* and *limited time* in the ad text works better than using *great*. Our key observation in the illustrated example is that although the advertisers are learning in isolation, the ad platform can learn across advertisers. In fact, most ad platforms are authorized to use performance data across advertisers in an *aggregate manner* to help advertisers perform better; however, using A/B test data across advertisers in a collaborative manner to automate ad creative refinement is a largely unexplored topic.

In this paper we address several sub-problems in ad creative (text and image) refinement exploiting the above observation by using multi-advertiser A/B test data:

(1) ad text generation: given an input ad creative, the task is to generate refined ad text,
(2) ad text keyphrase recommendation: given an input ad creative, the task is to recommend keyphrases for inclusion in the refined ad text, and
(3) ad image tag recommendation: given an input ad creative, the task is to recommend image tags (objects) to guide the selection of a refined ad image.

Another novelty in our proposed approaches for the above tasks is that they do not depend on intermediate models such as CTR prediction as required in previous work [5, 11] but rely on pairs of examples of the form: (low CTR creative, high CTR creative) where the CTR is based on the same population of users (*i.e.*, targeting is fixed). Both creatives in a pair are sourced from the same advertiser, and at a high level, the task of refining can be seen as *translating* the low CTR creative (source) to the high CTR creative (target). As we discuss in this paper, such pairs can be naturally collected from A/B tests conducted by multiple advertisers in an ad platform. Our main contributions are as follows.

- We solve three tasks around ad creative refinement: (i) ad text generation, (ii) keyphrase recommendation, and (iii) image tag recommendation.
- For ad text generation, we demonstrate that using a copy mechanism to selectively copy parts of the input ad text while introducing new words in the refined (generated) text is significantly better than baselines.
- For keyphrase and image tag recommendation, we demonstrate the efficacy of a deep relevance matching model for ranking keyphrases and image tags. We also show the relative robustness of keyphrase ranking (compared to text generation) in a cold-start scenario with unseen advertisers. We observed a 87% CTR increase via such recommendations for a major advertiser on Yahoo Gemini.

The remainder of the paper is organized as follows. Section 2 covers related work, and Section 3 covers problem formulation. Section 4 explains data sources, and creation of pairs of creatives for training ad refinement models. Section 5 covers proposed methods, Section 6 covers experimental results, and there is a discussion in Section 7.

2 RELATED WORK

2.1 Online advertising

Today, advertisers work with ad platforms [5, 19, 26] to launch campaigns that show ads to users on different websites. Advertisers design *one or more creatives* with the help of creative strategists to target relevant online users and measure the effectiveness of campaigns with metrics such as click-through-rate (CTR = $\frac{clicks}{impressions}$) are associated with the ad creative being tested. It is common for advertisers to do exploratory A/B tests with a large pool of creatives to efficiently learn which creative works best (popularly known as dynamic creative optimization in the industry) [16]. However, automatically understanding ad creatives (multi-modal in nature due to the presence of text and an image) and leveraging this understanding to create a pool of relevant creatives for A/B testing is emerging as an active area of research as described below. Understanding content in ad images and videos from a computer vision perspective was first studied in [12], where manual annotations were gathered from crowdsourced workers for: ad category, reasons to buy products advertised in the ad, and expected user response given the ad. Leveraging the dataset in this work, [20] studied recommending keywords for guiding a brand's creative design. However, [20] was limited to only text inputs for a brand (*e.g.*, the brand's Wikipedia page), and the recommendation was limited to single words (keywords). In [27], the setup in [20] was extended by including multi-modal information from past ad campaigns, *e.g.*, images, text in the image (OCR), and Wikipedia pages of associated brands. In this paper, we focus on refining existing (input) ad creatives, *i.e.*, the refinement is specific to the input ad creative as opposed to providing recommendations for an input

advertiser in [20, 27]. In addition, the usage of A/B test data across advertisers is another key difference with respect to prior work. Our approaches are limited to consuming only CTR data across advertisers (and not conversions), since in most cases, it is *owned* by the ad platform, which is typically authorized to use aggregated data across advertisers to make system-wide improvements (not biased towards a particular advertiser).

2.2 Relevance matching

One of our goals is to recommend a set of highly relevant keyphrases and image tags for improving an (input) ad creative. This can be modeled as a query-document relevance ranking problem [9], or as a collaborative filtering problem where *user-item* latent representations are used for recommendations [10, 14]. However, given the restriction on the number of keyphrases/image tags recommended, and their relevance to the advertiser under consideration, we focus on relevance ranking models (*e.g.*, DRMM [9] and variants [25]) in our keyphrase and image tag ranking setups.

2.3 Text-to-text generation

We formulate ad text generation as a sequence-to-sequence prediction task, which is common in natural language processing problems like machine translation and abstractive summarization. State-of-the-art performance in machine translation is typically obtained with an encoder-decoder neural architecture with attention [18]. In abstractive summarization, where both the source and target sequences are in the same language, an additional mechanism to copy input tokens to the output sequence has proven to be beneficial [24]. In the context of ad text generation, recent work [11] explored the use of an encoder-decoder architecture to automatically generate ad text based on an advertiser's webpage. The main differences between our work and [11] lie in: (i) studying ad refinement as opposed to generating an ad from scratch, (ii) the use of A/B test data across advertisers to train refinement models.

3 PROBLEM FORMULATION

We study three tasks around creative refinement as described below.

3.1 Task 1: ad text generation

In this task, the goal is to generate refined ad text (output) given an input ad (text and image). For example, considering the illustration in Figure 1 for advertiser #N, if the input ad text is '*great offers on cowboy boots!*', a possible generated output could be '*limited time offer on new cowboy boots!*'. We assume that the input ad image is retained for use with the output ad text. Additional metadata in the form of ad image (tags) and associated advertiser category is also assumed to be available. The output ad text is expected to have at least Δ% better CTR performance compared to the input ad text (where Δ is a design choice) and the output ad text is assumed to be targeted to the same population of users as the input ad.

3.2 Task 2: ad text keyphrase ranking

This is a simpler variant of task 1, where instead of generating the entire ad text, the task is to recommend keyphrases in the refined ad text. We formulate this as a ranking problem, where one needs to rank keyphrases from a given vocabulary, for inclusion in the refined ad text. For example, in Figure 1 for advertiser #N, if the input ad text is '*great offers on cowboy boots!*', a recommended list of keyphrases could have '*limited time*' and '*new*' as the top ranked keyphrases. The motivation here is to study cases when target text generation is hard to achieve, but useful keyphrase recommendations can still be provided. The objective is to recommend keyphrases that would increase the CTR if included in the ad text while keeping all other aspects of the ad (such as ad image) constant.

3.3 Task 3: ad image tag ranking

In this task, given an input ad image and text, the goal is to recommend image tags (output) to refine the ad image. Image tags essentially correspond to objects in the image, and are sourced from a given vocabulary of tags (explained later in Section 4.3). This task is the visual parallel of task 2, where instead of recommending textual keyphrases, we recommend tags for refining the ad image. The image tags can be used to select an ad image from a pool of images (*e.g.*, via a stock image library [2]); however, selecting or generating the final ad image is beyond the scope of this paper, and our study is limited to recommending image tags for the refined ad image. For example, in Figure 1 for advertiser #1, if the input ad text is the one with multimedia devices, a recommended list of image tags could contain '*human*' as a top ranked image tag. In addition, we assume that the input ad text is retained, and is available as metadata along with the associated advertiser category. The selection of an ad image based on the recommended tags is expected to increase the CTR of the refined creative.

4 DATA

In this section, we first explain the ad platform setup in Section 4.1; specifically Yahoo Gemini ad platform, however, the underlying hierarchical structure is fairly standard in the advertising industry. This is followed by our method for leveraging the ad platform setup to form ordered pairs of creatives (Section 4.2); the ordered pairs of creatives have a crucial role in our proposed methods to solve the tasks outlined in Section 3. In Section 4.3, we cover additional steps to automatically annotate the ad creative pairs with matched keyphrases and identified image tags. Finally in Section 4.4, we describe data insights which motivate our approaches.

4.1 Ad platform setup

As shown in Figure 2, an advertiser in the Yahoo Gemini ad platform can create multiple campaigns and each campaign can have multiple ad-groups. Each ad-group is tied to a pre-specified target audience. For example, if the advertiser is a major telecommunications company, different campaigns may represent different offerings from the company (*e.g.*, mobile phone plans and WiFi routers) whereas examples of ad-group targeting can be *seniors in New York City* and *males in San Francisco*. As shown in Figure 2, there can be multiple ad-ids in an ad-group; each ad-id has an ad text and image associated with it. For each qualifying user for the ad-group, one of the ad-ids is shown at random; in other words, if there is CTR performance difference across the ad-ids, it can be purely attributed to the differences in ad image and text across the ad-ids in the ad-group. For the example shown, the difference in CTRs of ad-id 1 and 2 can be attributed to the difference in the ad

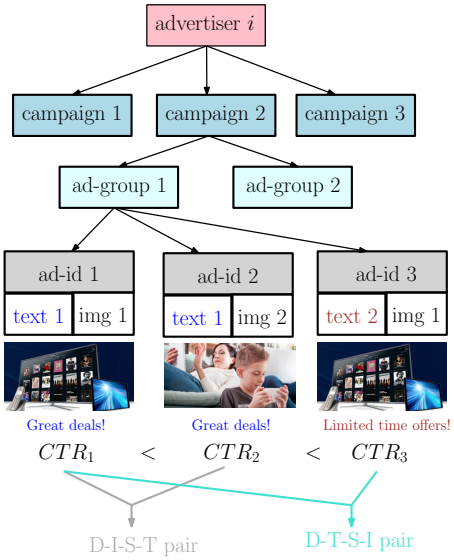

Figure 2: Ad campaign setup with multiple ad-groups and ad-ids. Difference in CTRs across ad-ids in the same ad-group can be attributed to differences in ad text and image. Ad-ids 1 and 2 form a different-image-same-text (D-I-S-T) pair, while 1 and 3 form a different-text-same-image (D-T-S-I) pair.

image, while for ad-ids 1 and 3, the difference can be attributed to the difference in ad text. However, in the case of ad-ids 2 and 3, the CTR difference is a result of differences in both the image and text.

4.2 Constructing ad creative pairs

We use data from ad-groups across multiple advertisers to form two datasets: (i) different-text-same-image (D-T-S-I) dataset, and (ii) different-image-same-text (D-I-S-T) dataset as described below.

4.2.1 D-T-S-I dataset. To create this dataset, from each ad-group, we create pairs of ad-ids (creatives) such that in each pair the ad text is different but the ad image is same. Furthermore, in each such pair, we order the ad-ids as (source, target) where source CTR is lower than target CTR. For example, in Figure 2, (ad-id 1, ad-id 3) form such a (source, target) pair in the D-T-S-I dataset. We collect such pairs using ad-groups across multiple advertisers. In case multiple pairs have the same source ad text (but different target ad text), we only retain the pair with highest CTR difference, and discard the other (duplicate-source) pairs. Finally, we keep the pairs where the relative CTR difference is higher than $\Delta\%$ (design choice). The intuition behind creating such pairs is to provide training examples to an ad text refinement model, *e.g.*, for generating the target ad text given the source ad text (explained in Section 5.1).

4.2.2 D-I-S-T dataset. To create this dataset, from each ad-group, we create pairs of ad-ids such that the ad image is different but the ad text is same. As in the D-T-S-I dataset, we order the ad-ids in the pair as (source, target) where source CTR is lower than target CTR; in Figure 2, (ad-id 1, ad-id 2) is an example of such a pair. We collect such pairs across ad-groups of multiple advertisers. If there are pairs with the same source image, we retain the pair with the highest CTR difference and discard the other duplicates. Finally, we filter out pairs with relative CTR difference below $\Delta\%$ (design choice). The intuition behind creating such pairs is to provide training examples for refined ad images given source ad text and image.

4.3 Keyphrases and image tags annotation

For each pair in the D-T-S-I and D-I-S-T datasets, we add metadata in the form of matched keyphrases and image tags (explained below).

Keyphrases. We first form a vocabulary of keyphrases using an unsupervised keyphrase extraction method[1] on the collective ad text corpus (including both source and target ad text from all pairs). For example, from retail advertisers, typical examples of extracted keyphrases include phrases like *free shipping* and *limited time offers*, while from telecommunication advertisers, examples include *high speed internet* and *bundle deals*. Using the obtained vocabulary of keyphrases, for each pair in the D-T-S-I and D-S-T-I datasets, we add a list of exact matches found in the source and target ad text.

Image tags. Image tags are the objects detected in an image via the (pre-trained) Inception Resnet v2 object detection model as in the Open Images V2 repository [15]. We extract these image tags from the source and target ad images in D-T-S-I and D-I-S-T datasets. Inception Resnet v2 [15] is a convolutional neural network trained by Google on Flickr images in the Open Images V2 dataset. It has about 5000 classes (possible tags in an image). Each image can have multiple tags and the model returns a list of inferred tags with confidence scores. We retain all tags with a score above 0.8. For example, the ad image in ad-id 2 in Figure 2 has tags *woman, child, face*, whereas the image in ad-id 1 has the tag *multimedia*.

4.4 Insights from D-I-S-T and D-T-S-I datasets

Based on 5 months (July–November 2019) of data from the Yahoo Gemini platform, we gathered several insights from D-T-S-I and D-I-S-T datasets spanning a sample of over 3500 advertisers. The minimum CTR difference (Δ) in each source-target pair was kept at 10%. We highlight key insights below which guided our proposed approaches (additional statistics are covered later in Section 6).

High word overlap between source and target text. In the D-T-S-I dataset, the average number of words in both target and source ad text is close to 13 (sequence length), but there is a 60% overlap between words in source and target. This indicates: (i) target retains a lot of words from the source (plausibly to preserve context), and (ii) there are word replacements in source to keep the sequence length roughly the same. Hence, a *copy mechanism* which can selectively copy parts of the source text while introducing new words in target looks intuitive for the ad text generation task (details in Section 5.1).

Discriminative power of keyphrases and image tags. An advertiser category-wise case study using the D-T-S-I dataset revealed that the presence of certain keyphrases in the target ad text (and their absence in the source) consistently led to higher CTR relative to the source. For example, in the case of retail category advertisers, such keyphrases included *free shipping* and *limited time offer*. In a parallel

[1]We used multipartite-rank [7] method implemented in the PKE keyphrase extraction package [6]. Choice of this method (versus others in PKE, *e.g.*, TF-IDF, and Position-rank [8]) was guided by visual inspection of results on representative advertisers.

study using the D-I-S-T dataset, we observed analogous results with image tags. For example, for telecommunication advertisers, we found that target images with human elements (*i.e.* having tags *woman*, *man*, *child*) had higher CTR than source images with just *multimedia* tag. The above insights motivate the use of a ranking approach for recommending keyphrases and image tags for refining an input ad creative (details in Section 5.2).

5 GENERATION AND RANKING MODELS

We now describe, the proposed solutions for tasks 1-3 (Section 3). The text generation approach for task 1 (Section 3.1) is explained in Section 5.1. For tasks 2 (Section 3.2) and 3 (Section 3.3), the proposed keyphrase/image tag ranking model is explained in Section 5.2.

5.1 Ad text generation model for task 1

Task 1 can be formulated a sequence-to-sequence (seq2seq) prediction task, where given an input ad text (source sequence of tokens), the predicted output should be a refined version of the input ad text (target sequence) with a higher expected CTR. The construction of the D-T-S-I dataset (in Section 4.2.1) is naturally suited for training such a seq2seq model, since in each pair the target ad text has higher CTR than the source ad text (the same ad-group, and same image constraints in each pair eliminate all other confounding factors affecting CTR). Given the D-T-S-I dataset, to solve task 1, we propose using an encoder-decoder architecture with a mechanism to selectively *copy* words from the source text; the intuition behind the proposal, and underlying architecture details are explained below.

Intuition. We borrow ideas from state-of-the-art models in abstractive summarization [24] and use it to solve task 1 as follows. We use an encoder-decoder architecture with attention [4], along with a copy mechanism [24] as shown in Figure 3. In our setup, the

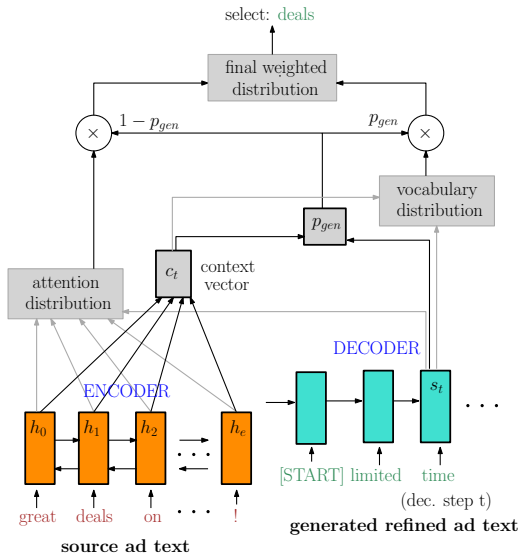

Figure 3: Encoder-decoder with attention and copy mechanism for generating refined (target) ad text given source ad.

motivation for using the copy mechanism is driven by the observation that there is a 60% overlap between source and target words in the D-T-S-I dataset (as mentioned in Section 4.4). It is plausible that copying some words from the source is good enough to preserve the underlying context, while adding new words in the target can boost the CTR. We describe the underlying model details below.

Model details: We use a bidirectional LSTM encoder for the source sequence and an LSTM decoder for the target sequence [4]. Following [18], the attention distribution is computed as:

$$e_i^t = h_i^\top W_{\text{att}} s_t, \quad a^t = \text{softmax}\left(e^t\right), \tag{1}$$

where h_i is the encoder hidden state, s_t is decoder state at step t, a^t is the attention distribution, and W_{att} represents the learnable parameters. The attention-weighted sum of all encoder hidden states is used compute the context vector as:

$$c_t = \sum_i a_i^t h_i. \tag{2}$$

The generation probability p_{gen} for step t is computed using the context vector (c_t), decoder state (s_t) and decoder input (x_t) as:

$$p_{\text{gen}} = \sigma\left(w_c^\top c_t + w_s^\top s_t + w_x^\top x_t + b_{\text{ptr}}\right), \tag{3}$$

where w_c, w_s, w_x are vectors and b_{ptr} is a scalar, all of which are learnable; $\sigma(\cdot)$ denotes the sigmoid function. Here, p_{gen} is used to *softly* choose between generating a word from the entire vocabulary versus copying a word (token) from the input sequence (via sampling from the attention distribution a^t). The vocabulary distribution for generating a new word can be computed as:

$$\mathbb{P}_{\text{vocab}} = \text{softmax}\left(V'(V\left[s_t; c_t\right] + b) + b'\right), \tag{4}$$

where V, V', b, and b' are learnable parameters. With p_{gen}, the effective distribution over the vocabulary can be written as:

$$\mathbb{P}(y) = p_{\text{gen}}\mathbb{P}_{\text{vocab}}(y) + (1 - p_{\text{gen}}) \sum_{i:y_i=y} a_i^t, \tag{5}$$

where y is a word in the vocabulary. For training, the loss at step t ($\mathcal{L}_t$) is the negative log-likelihood associated with target word y_t^*, and that of the whole sequence is simply the average:

$$\mathcal{L}_t = -\log\left(\mathbb{P}(y_t^*)\right), \quad \mathcal{L} = \frac{1}{T}\sum_{t=0}^{T} \mathcal{L}_t. \tag{6}$$

Our implementation of the above model leveraged OpenNMT-Py [13] with: train steps = 200k, optimizer = SGD, and batch size = 128.

5.2 Ranking model for tasks 2 and 3

We consider solving the keyphrase (and image tag) recommendation problem via a ranking model, where the model outputs a list of keyphrases (and image tags) in decreasing order of relevance for a given ad creative. We describe below the model for the keyphrase ranking task; the image tag ranking model is analogous, and we skip its description for brevity. We use the state-of-the-art pairwise deep relevance matching model (DRMM) [9, 25] whose architecture for our recommendation setup is shown in Figure 4. It is worth noting that our pairwise ranking formulation can be changed to accommodate other multi-objective or list-based loss-functions. We chose the DRMM model since it is not restricted by the length of input, as most ranking models are, but relies on capturing local

interactions between query and document terms. Given a *(source ad text , target keyphrase)* combination, the model first computes the top-k interactions between the source ad text words and the keyphrases. These interactions are passed through a multi-layer perceptron (MLP), and the overall score is aggregated with a query term gate which is a softmax function over all terms in that query. DRMM employs a pair-wise ranking loss function as described

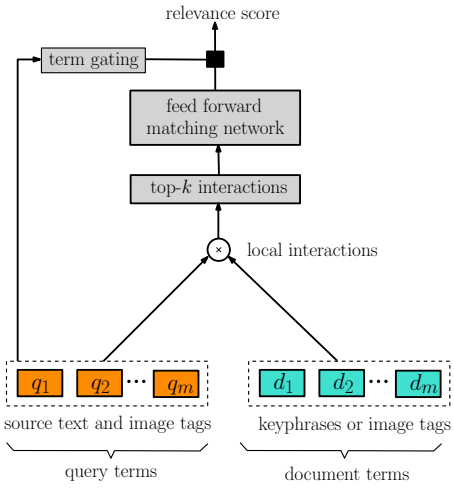

Figure 4: DRMM-(top k) for keyphrase/image tag ranking.

below. We denote the source ad text by just src in the following explanation. Given a triple (src, p^+, p^-) where keyphrase p^+ is ranked higher than keyphrase p^- with respect to src, the loss function is:

$$\mathcal{L}(src, p^+, p^-; \theta) = max(0, 1 - s(src, p^+) + s(src, p^-)), \quad (7)$$

where $s(src, p)$ denotes the predicted matching score for keyphrase p, and the source ad text. Metadata in the form of image tags, and advertiser category can be introduced as additional query terms. In our implementation, we used the top-k version of DRMM [25] in Match-Zoo [1] with $k = 20$ and ADAM optimizer.

6 RESULTS

In this section, we first cover notable statistics of the D-T-S-I and D-I-S-T datasets in Section 6.1, followed by a description of evaluation metrics in Section 6.2. This is followed by results on ad text generation, keyphrase ranking, and image tag ranking.

task	keyphrase ranking		image-tag ranking	
	source	target	source	target
vanilla-split				
# tokens (1)	12.24±3.6	12.25±3.6	13.44±3.8	13.44±3.8
# tokens (2)	12.37±3.6	12.32±3.6	13.44±3.7	13.44±3.7
# kp/img (1)		12.02±4		7.14±3.8
# kp/img (2)		12.13±4		7.06 ±3.8

Table 1: Mean (±std) of attributes of train (1) and test (2) sets: number of words in ad text (# tokens), number of matched keyphrases (# kp), and number of image tags (# img).

6.1 Dataset statistics

The D-T-S-I and D-I-S-T datasets were built using a sample of 5 months of data from Yahoo Gemini (July-November 2019). The data consisted of over 3500 advertisers (> 8500 campaigns in English for U.S. audiences, ∼ 100 categories), and each ad-id considered in the dataset had over 10, 000 impressions. After the filtering process (*i.e.*, keeping only source target pairs with more than $\Delta = 10\%$ CTR difference, and removing duplicate sources), the D-T-S-I dataset consisted of over 20, 000 pairs while the D-I-S-T dataset consisted of over 10, 000 pairs. Each dataset was randomly divided into train, test and validation sets in proportions of 77%, 20%, and 3% respectively; we will refer to this as a *vanilla* split. In addition to the vanilla split, we created a cold-start split where there was no overlap between advertisers in train, test and validation sets; this presents a much more difficult (versus vanilla split) learning problem with unseen advertisers Additional dataset statistics are shown in Table 1 .

6.2 Evaluation metrics

For ad text generation, we use standard metrics for text generation problems: (i) BLEU [21], and (ii) ROUGE scores [17]. We introduce metrics to gauge the presence of matched (target) keyphrases in the generated sequence: (i) keyphrase-precision (kp-P), (ii) keyphrase-recall (kp-R), and (iii) keyphrase -F (kp-F). In other words, we compute precision and recall for target keyphrases, considering the list of tokens in the generated text. For both keyphrase and image tag ranking, we use: (i) precision at k ($P@k$), (ii) recall at k ($R@k$), and (iii) normalized cumulative discounted gain at k ($NDCG@k$).

6.3 Ad text generation results

Table 2 covers generation results for the vanilla, and cold-start cases (metrics on test set). The baseline scores are for the case when the source ad text is considered as the predicted ad text (*i.e.*, no change in input), and compared with the target ad text. In Table 2, CAT and IMG denote the addition of category and image tags to beginning of the input sequence (image tags in alphabetical order). The main observations are as follows.

Copy mechanism works. In both vanilla and cold-start cases, there is a significant lift in the metrics due to the copy mechanism. In case of vanilla split, the copy mechanism is able to beat the baseline (predicted sequence = source sequence) metrics. However, in cold-start, it is below the baseline (but is better than the no-copy version).

Category helps. There is a consistent improvement in metrics on using category metadata in the input sequence. As expected, category information provides a relatively higher lift (4.4% above ATTN+CP in ROUGE-L F) for cold-start split compared to vanilla split (0.2% lift). In comparison, adding image tags to the input sequence (along with category) does not provide any lift (suggesting the need for better ways to incorporate image information).

Cold-start is challenging. We computed the histogram of ROUGE-L F scores on the test set using the best model (ATTN+CAT+CP) for the vanilla-split (Figure 5) and cold-start split (Figure 6) cases; for both splits, the baseline ROUGE-L F is around 61. As shown, for vanilla, the distribution has a significant number examples above the baseline, while the distribution's mass significantly shifts

model	BLEU	ROUGE-1 F	ROUGE-2 F	ROUGE-L F	kp-P	kp-R	kp-F
vanilla-split							
baseline (pred=src)	56.28	63.49	50.79	61.13	0.643	0.644	0.643
ATTN	50.74	57.62	47.26	56.01	0.552	0.548	0.55
ATTN + CP	59.38	65.61	55.13	63.79	0.661	0.648	0.655
ATTN + CP + CAT	**59.45**	**65.74**	**55.35**	**63.91**	**0.661**	**0.649**	**0.655**
ATTN + CP + CAT + IMG	58.37	65.63	55.18	63.82	0.663	0.646	0.654
cold-start split							
baseline (pred=src)	56.01	63.69	51.02	61.57	0.643	0.637	0.64
ATTN	16	26.64	13.29	25.02	0.195	0.177	0.185
ATTN + CP	34.39	45.26	30.86	42.81	0.462	0.422	0.441
ATTN + CP + CAT	**35.91**	**47.52**	**32.64**	**44.69**	**0.494**	**0.434**	**0.462**
ATTN + CP + CAT + IMG	33.42	44.33	29.53	41.76	0.422	0.37	0.394

Table 2: Ad text generation results: ATTN denotes the LSTM encoder-decoder with attention model, CP denotes copy mechanism, CAT denotes adding category, and IMG denotes adding source image tags.

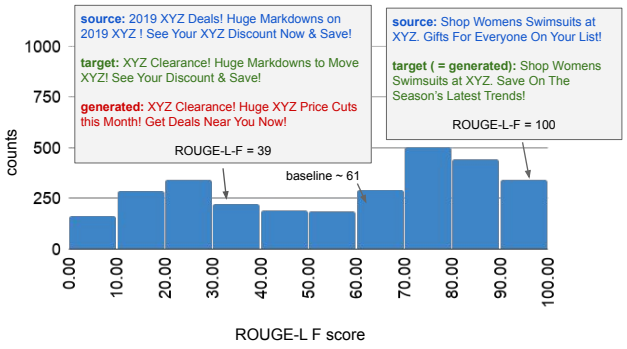

Figure 5: Histogram of ROUGE-L F scores in test set for vanilla-split (ATTN + CP + CAT model). Two anonymized examples are also shown with their ROUGE-L F scores.

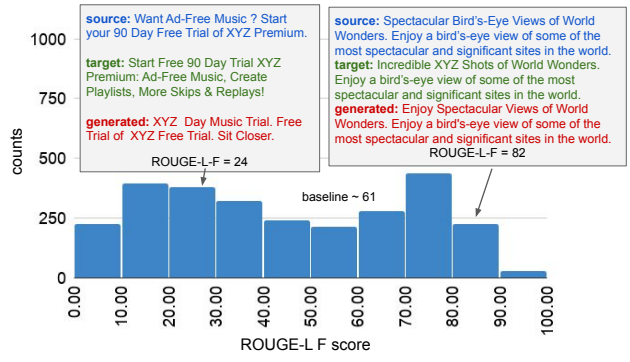

Figure 6: Histogram of ROUGE-L F scores in test set for cold-start split (ATTN + CP + CAT model). Two anonymized examples are also shown with their ROUGE-L F scores.

below baseline for cold-start. The listed examples of generated text give a sense of how good the generated outputs are in terms of human judgement vis-a-vis ROUGE-L F scores. The keyphrase based metrics for generation (kp-P,R,F) as reported in Table 2, are helpful in gauging the extent to which target keyphrases appear in the generated text. For example, in the lower scored generated text in Figure 5, *clearance* is correctly introduced, but *price cuts* is incorrectly introduced. Although *price cuts* is incorrect given the target text (which has a guaranteed CTR lift), it remains to be seen if it leads to a lower CTR online (beyond the scope of this paper).

6.4 Keyphrase ranking results

Table 3 shows the results for keyphrase ranking. The baselines included methods using: (i) cosine similarity (EMB-SIM) based on Glove [22] embeddings for keyphrases and input text (average of word embeddings), and (ii) TF-IDF representation of source ad text and keyphrases is used to compute similarity and keyphrases are ranked in descending order of similarity. As shown in Table 3, for

model	P5	P10	R5	R10	ndcg5	ndcg10
vanilla						
EMB-SIM	0.17	0.10	0.07	0.09	0.19	0.14
TF-IDF	0.33	0.26	0.15	0.23	0.35	0.30
DRMM	0.50	0.39	0.25	0.38	0.53	0.47
+ CAT	**0.51**	**0.40**	**0.25**	**0.39**	**0.53**	**0.48**
+ CAT + IMG	0.41	0.32	0.21	0.32	0.43	0.39
cold st.						
EMB-SIM	0.12	0.07	0.05	0.06	0.14	0.11
TF-IDF	0.27	0.21	0.12	0.18	0.29	0.26
DRMM	0.38	0.29	0.22	0.32	0.41	0.37
+ CAT	**0.42**	**0.32**	**0.24**	**0.36**	**0.45**	**0.40**
+ CAT + IMG	0.34	0.26	0.20	0.30	0.36	0.33

Table 3: Kephrase ranking: baselines versus DRMM, and the effect of adding category and image tags as query terms.

both splits, using DRMM with category features performs the best in terms of all metrics. Cold-start best performance is comparable to the vanilla split best performance (*e.g.*, 7% drop in $R@10$, compared to 33% drop in $kp - R$ in Table 2 for generation). Hence,

split	metric	add-0	add-1	add-2	add-3	add-10
cold-start	kp-P	0.50	0.50	0.49	0.46	0.35
cold-start	kp-R	0.43	0.45	0.46	0.47	0.53

Table 4: Ranking-aided keyphrase metrics for generation. Add-0 denotes no assistance, and add-10 denotes adding top 10 ranked keyphrases in the generation output.

source ad text	target ad text	generated ad text	recommended keyphrases
Transform Your Workout For Better Results! The all new XYZ with workouts is what you need to make the best of your workout.	Home Fitness Equipment. XYZ are the ellipticals you need to get the right motivation to stay fit.	Shop The Official new XYZ from XYZ from XYZ. Free Shipping!	{'workout', 'fitness', 'schedule', 'real workout' }
Biggest List of Senior Discounts. Click and grab the most popular Senior Discounts available.	Savings Trick Every Senior Should Know About. Incredible Offers Seniors Are Taking Advantage.	Biggest Senior Discounts. Discounts the most popular Senior Discounts available.	{'senior', 'biggest discounts', 'offers ', 'compare '}

Figure 7: Sample keyphrase ranking results vis-a-vis generated ad text (cold-start); the keyphrase recommendations can cover target keyphrases missed by poor generation.

keyphrase ranking seems to be more *robust* to unseen advertisers compared to ad text generation. As seen in text generation, naively adding image tags to the input along with category does not generalize well (mildly hurts performance). We suspect that since image tags represent objects, they provide no additional context for the ranker to select better keyphrases. Most often, keyphrases provide more information about the brand, and image tags that represent objects may not add any complementary information about the brand directly that the ranking model can exploit. In future, we shall explore features that encapsulate information in the image directly [27] rather than use image tags for keyphrase ranking. We also study the possibility of *assisting* generation results with corresponding ranking results. Table 4 shows the boost in kp-R for the best generation results (ATTN+CP+CAT), when the corresponding (top-r) outputs of the DRMM + CAT model are added to the list of matched keyphrases in generated text. As shown for cold-start, just adding the top ranked keyphrase (add-1) improves the recall ($0.43 \rightarrow 0.45$) without affecting the precision (0.5). This indicates that ranking results can complement generation results in a helpful manner (illustrative cold-start examples in Figure 7).

6.5 Ad image tag ranking results

Table 5 shows the ranking results for image tags (baselines, and CAT + IMG feature additions in DRMM). Using DRMM with category and image tags performs the best. The efficacy of image tags (in source) to predict relevant tags (in target) may be linked to common modality. We also report the frequent top ranked image tags for selected categories in Table 6 using the DRMM + CAT + IMG model.

6.6 Online results

We deployed the ranking models for tasks 2 and 3 (*i.e.*, keyphrase and image tag rankers) as an internal service for Yahoo Gemini account teams which manage campaigns of major advertisers. To study end-to-end adoption, we partnered with the account team for an Internet service provider. Using their existing creative (text and image) as input, the top keyphrase and top image tag recommendation were considered. The advertiser approved an A/B test for the refined creative (incorporating both image and text refinements together) versus their existing creative. The A/B test was conducted for 2 weeks via Yahoo Gemini, and the refined creative showed an 87% improvement in CTR, validating the model recommendations.

model	P5	P10	R5	R10	ndcg5	ndcg10
vanilla						
EMB-SIM	0.16	0.15	0.12	0.22	0.18	0.23
TF-IDF	0.27	0.24	0.21	0.35	0.29	0.37
DRMM	0.49	0.34	0.35	0.49	0.53	0.50
+ CAT	0.50	0.35	0.36	0.49	0.54	0.51
+ CAT + IMG	**0.51**	**0.37**	0.34	**0.49**	**0.55**	**0.52**
cold st.						
EMB-SIM	0.16	0.15	0.11	0.20	0.18	0.22
TF-IDF	0.28	0.24	0.20	0.33	0.29	0.36
DRMM	0.41	0.31	0.29	0.43	0.44	0.44
+ CAT	0.43	0.33	0.31	0.45	0.46	0.46
+ CAT + IMG	**0.53**	**0.37**	**0.38**	**0.52**	**0.58**	**0.55**

Table 5: Image tag ranking: baselines versus DRMM, and the effect of adding category and image tags as query terms.

category	top 5 ranked image tags
apparel	clothing, face, hair, girl, pattern
job portals	face, clothing, multimedia, road, man, woman
auto	wheel, car, motorcycle, clothing, face
real estate	man, woman, mansion, bedroom, kitchen

Table 6: Frequent top ranked image tags by category.

7 DISCUSSION

Our results show the efficacy of using A/B test data across advertisers for both generation and ranking formulations of ad creative refinement. Account teams testing the proposed models requested additional evidence in the form of CTR of similar ads (*i.e.*, with recommended keyphrases and image tags) to convince advertisers to approve tests for refined creatives. Studying the extent of adoption by advertisers and using this feedback to control creative generation is a promising direction for future research.

REFERENCES

[1] [n.d.]. Match Zoo. https://github.com/NTMC-Community/MatchZoo.
[2] [n.d.]. Shutterstock: Search millions of royalty free stock images, photos, videos, and music. https://www.shutterstock.com/.
[3] [n.d.]. Taboola-trends. https://trends.taboola.com/.
[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. [n.d.]. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR 2015*.
[5] Narayan Bhamidipati, Ravi Kant, and Shaunak Mishra. [n.d.]. A Large Scale Prediction Engine for App Install Clicks and Conversions. In *CIKM 2017*.
[6] Florian Boudin. 2016. pke: an open source python-based keyphrase extraction toolkit. In *COLING 2016*.
[7] Florian Boudin. 2018. Unsupervised Keyphrase Extraction with Multipartite Graphs. In *ACL 2018*.
[8] Corina Florescu and Cornelia Caragea. 2017. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In *ACL 2017*.
[9] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM 2016*.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*.

[11] J. Weston Hughes, Keng-hao Chang, and Ruofei Zhang. [n.d.]. Generating Better Search Engine Text Advertisements with Deep Reinforcement Learning *(KDD âĂŹ19)*.

[12] Zaeem Hussain, Mingda Zhang, Xiaozhong Zhang, Keren Ye, Christopher Thomas, Zuha Agha, Nathan Ong, and Adriana Kovashka. 2017. Automatic Understanding of Image and Video Advertisements. In *CVPR*.

[13] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT. In *ACL 2017*.

[14] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *KDD*.

[15] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. 2017. OpenImages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages* (2017).

[16] Wei Li, Xuerui Wang, Ruofei Zhang, Ying Cui, Jianchang Mao, and Rong Jin. 2010. Exploitation and Exploration in a Performance Based Contextual Advertising System. In *KDD 2010*.

[17] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. ACL.

[18] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP 2015*.

[19] H. Brendan McMahan, Gary Holt, D. Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, and Jeremy Kubica. [n.d.]. Ad Click Prediction: a View from the Trenches *(KDD 2013)*.

[20] Shaunak Mishra, Manisha Verma, and Jelena Gligorijevic. 2019. Guiding Creative Design in Online Advertising *(RecSys 2019)*.

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. ACL.

[22] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP 2014*.

[23] Susanne Schmidt and Martin Eisend. 2015. Advertising Repetition: A Meta-Analysis on Effective Frequency in Advertising. *Journal of Advertising* (2015).

[24] Abigail See, Peter J. Liu, and Christopher D. Manning. [n.d.]. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL 2017*.

[25] Zhou Yang, Qingfeng Lan, Jiafeng Guo, Yixing Fan, Xiaofei Zhu, Yanyan Lan, Yue Wang, and Xueqi Cheng. [n.d.]. A Deep Top-K Relevance Matching Model for Ad-hoc Retrieval. In *Information Retrieval - 24th China Conference, CCIR 2018*.

[26] Yichao Zhou, Shaunak Mishra, Jelena Gligorijevic, Tarun Bhatia, and Narayan Bhamidipati. 2019. Understanding Consumer Journey using Attention based Recurrent Neural Networks. *KDD* (2019).

[27] Yichao Zhou, Shaunak Mishra, Manisha Verma, Narayan Bhamidipati, and Wei Wang. [n.d.]. Recommending Themes for Ad Creative Design via Visual-Linguistic Representations *(The Web Conference 2020)*.