Don't Miss the Potential Customers!
Retrieving Similar Ads to Improve User Targeting

Yi Feng[1], Ting Wang[3], Chuanyi Li[1], Vincent Ng[2],
Jidong Ge[1], Bin Luo[1], Yucheng Hu[3] and Xiaopeng Zhang[3]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Human Language Technology Research Institute, University of Texas at Dallas, USA
[3]Tencent, China
fy@smail.nju.edu.cn, lcy@nju.edu.cn, vince@hlt.utdallas.edu

Abstract

User targeting is an essential task in the modern advertising industry: given a package of ads for a particular category of products (e.g., green tea), identify the online users to whom the ad package should be targeted. A (ad package specific) user targeting model is typically trained using historical clickthrough data: positive instances correspond to users who have clicked on an ad in the package before, whereas negative instances correspond to users who have not clicked on any ads in the package that were displayed to them. Collecting a sufficient amount of positive training data for training an accurate user targeting model, however, is by no means trivial. This paper proposes a novel method for automatic augmentation of the set of positive training instances. Experimental results on two datasets, including a real-world company dataset, demonstrate the effectiveness of our proposed method.

1 Introduction

User targeting is an essential task in the e-commerce advertising industry. Informally, the goal of user targeting is to identify online users to whom a particular ad or ad *package* (i.e., a set of ads on a particular kind of products, such as green tea) should be targeted. Figure 1 shows a pipeline through which the user targeting task is typically tackled. Given an ad package that a company seeks to advertise, the company starts by randomly sampling a group of users from its customer database and displaying select ads in the package on the webpage(s) they visit. These users are then divided into two groups: *clicking* users and *non-clicking* users. Clicking users are those who clicked on the ads and therefore expressed interest in them, while their non-clicking counterparts are those who did not click on the ads and are presumably not interested in the ads. These two groups of users then serve as positive and negative examples for training a user targeting *model*, which can then be used to predict whether a new user should be targeted for the given ad package.

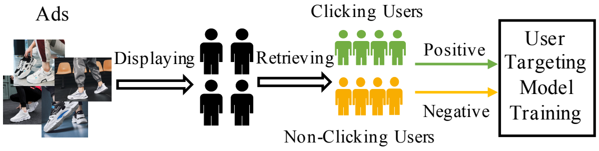


Figure 1: The user targeting pipeline.

While this approach of using historical clickthrough data to automatically collect data for training a user targeting model is appealing at first glance, it has a key weakness: it may take time to collect enough data to train a reliable user targeting model, especially for *long-tail* ads (i.e., ads with few or no clicks). Worse still, even after waiting long enough, we still cannot guarantee that there will be enough clicks to generate *positive* training examples. Collecting sufficient positive training examples is critical to the success of this approach.

To address this challenge, we put forward the following hypothesis: users who clicked on an ad for a particular product category (e.g., green tea) in the past are more likely to click on an ad for the same product category in the future. Given this hypothesis, we can potentially expedite the collection of positive examples for training a user targeting model as follows. Given a package of ads for a particular product category, we first identify ads for the same product category and then use their clicking users to augment the training data for training the user targeting model.

The question, then, is: how can we automatically identify ads for the same product category as the one under consideration? One approach would be to train a classifier to classify an ad according to its product category. While this approach is straightforward, the resulting classifier will fail to classify an ad for a (new) product category that is not seen in the training data, In light of this

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1493–1503
November 7–11, 2021. ©2021 Association for Computational Linguistics

weakness, we instead propose to learn how likely two ads are for the same product category. Not only will this address the aforementioned question of identifying ads for the same product category as the one under consideration, but the resulting model will be applicable to new product categories.

The next question is: how can we train a model to determine how likely two ads are for the same product category? Since ads are displayed in the form of creatives that are typically composed of both texts and images, a reasonable solution to this problem should involve matching the texts and the images in the two ads. While algorithms for text matching (Yang et al., 2019; Gong et al., 2018; Wang et al., 2017b), image matching (Schroff et al., 2015; Novotný et al., 2017), and text-image matching (Zheng et al., 2020; Wang et al., 2019) exist, none of them was developed specifically for ads. We therefore propose ION, a bimodal method that determines how likely two ads belong to the same product category, the key highlights of which include the design of (1) a semantics-enhanced image region extraction mechanism for identifying the region(s) of the image in an ad that is most relevant to the text, and (2) a dual-path fusion attention method for fusing the information extracted from the two modalities.

In sum, our contributions in this paper are three fold. First, we hypothesize that users who clicked on an ad belonging to a particular product category in the past is more likely to click on an ad belonging to the same category in the future, and exploit this hypothesis to augment the positive instances used to train a user targeting model. Second, we propose ION, a method for determining how likely two ads belong to the same product category, as a means to identify positive instances for user targeting. Finally, we evaluate ION in terms of (1) its effectiveness in retrieving ads with the same product category and (2) its ability to improve a user targeting model via augmenting the training set using the positive instances it identifies. Experiments on two datasets demonstrate its superiority to six baseline systems, providing suggestive evidence of its usefulness for the user targeting task.

The rest of this paper is structured as follows. Section 2 describes related work. In Section 3, we present ION, our model for determining how likely two ads belong to the same product category. Section 4 compares ION with state-of-the-art baselines on two datasets. Finally, we conclude in Section 5.

2 Related Work

Works related to user targeting exist. Unlike ours, they primarily focus on designing fancy models that are trained on a large amount of data (Zhou et al., 2018; Covington et al., 2016; Wang et al., 2017a). In contrast, we aim at solving the insufficient training data problem, which to our knowledge is an unexplored area of research.

A crucial aspect of our work concerns the development of a method for determining how likely two ads belong to the same product category. Below we will discuss related work on text matching, image matching, and text-image matching, even though none of the existing matching algorithms are specifically developed for ad matching.

Many **text matching** methods use an encoder such as RNNs (Bowman et al., 2015), CNNs (Tan et al., 2016), recursive networks (Tai et al., 2015) and Transformer-based networks (Vaswani et al., 2017; Devlin et al., 2019) to embed input texts into vectors, possibly enhanced by attention (Parikh et al., 2016; Chen et al., 2017), and then build a binary classifier to determine whether the inputs are similar. An exception is Yang et al. (2019), whose matching method is based on rich alignment features. In general, however, the text in ads are often so ambiguous that it is difficult to determine which products are promoted.

As for **image matching**, existing geometric feature detectors and descriptors can compute the similarity between images (e.g., Lowe (2004), Wang et al. (2018)), and a matching mechanism based on CNNs has been proposed to retrieval face images (Schroff et al., 2015). However, a large portion of an ad image usually contain background objects, which make the extracted image features too noisy to accurately determine the underlying products being promoted.

To perform **text-image matching**, some methods embed different modalities (e.g., texts and images) of the input into the same space and compute similarity from feature vectors (Wang et al., 2016; Zheng et al., 2020; Collell et al., 2017), but they may be too coarse-grained to exploit local features, i.e., words and image regions. Recent work (Karpathy and Li, 2015; Huang et al., 2018; Qi et al., 2018; Hu et al., 2019) split texts and images into fine-grained words and visual regions, and computes similarity by aligning the features of word semantics and those extracted from image regions, possibly with the help of attention (Lee et al., 2018)

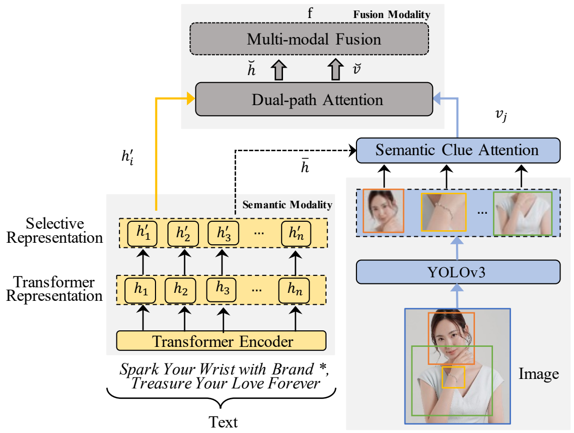


Figure 2: The framework of the proposed model.

and external knowledge (Shi et al., 2019; Wang et al., 2019). Different from work on cross-modal matching, which measures the similarity between different modalities, our work focuses on fusing features from the texts and the image in an ad to create a multimodal representation. Note that there is also related work that aims to generate multimodal vectors containing both text and image features for pre-training or classification (Xu et al., 2020; Abavisani et al., 2020; Lu et al., 2019), in which vectors from different modalities are concatenated to form the multimodal representation. Rather than performing a simple concatenation, our work proposes an attention mechanism to fuse modalities in order to better identify the correspondence between words and image regions. In addition, while existing methods do not determine which words and image regions in an ad are relevant to the product under consideration and which ones are irrelevant/noisy, our method encodes words and extract image regions selectively so that those that are related to the product are given larger weights.

3 Method

In this section, we describe our two-step method for determining how likely two ads belong to the same product category. During training, we train a model for learning an ad *representation* such that two ads that belong to the same product category have similar representations. After training, we can apply the resulting model as follows. Given an ad in the test set, we retrieve the k ads that are most similar to it, where similarity is computed using a similarity metric applied to the representations of two ads. The rest of this section focuses on the first step, in which we train the model using multi-task learning to learn ad representations (the main task) simultaneously with keyword extraction from text (the auxiliary task).

The model architecture is shown in Figure 2. Given an ad composed of text and an image as input, the model first embeds the sequence of words using Transformer (Section 3.1). After that, a Keyword-guided Selective Gate (KSG) mechanism is adopted to mine the semantics from these text representations (Section 3.2), which are leveraged as clues for an attention module that reranks the generated image regions extracted by the YOLOv3 object detection module (Redmon and Farhadi, 2018) (Section 3.3). Finally, the model combines the re-ranked image regions and the distributed text representation through a Dual-path Fusion Attention (DFA) layer to obtain a multimodal representation of the ad (Section 3.4). Below we introduce each of these modules in detail.

3.1 Sentence Representation Learning

We encode each word in the text portion of the input ad using Transformer (Vaswani et al., 2017), as it has been shown effective in many NLP tasks (Devlin et al., 2019; Liu and Lapata, 2019). Given the text, we encode its word sequence and obtain its representations $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \dots \mathbf{h}_n\}$, where n is the number of words and $\mathbf{h}_i \in \mathbf{R}^{d_{model}}$.

3.2 Keywords-Guided Selective Encoding

Some words in the text portion of an ad contain information that can help us to determine which products are promoted by the ads, and thus are more useful than those words that do not. As an example, the ad shown in Figure 2 contains the text "Spark Wrist with Brand XXX[1], Treasure Your Love Forever". Here, the words "Spark" and "Wrist" strongly suggest that it may be an ad of something that is sparkling and worn on the wrist. Furthermore, the brand may also indicate the product category that helps us to determine the ad product, as a brand advertiser usually sells products of only a small number of categories. Based on word semantics, it is highly likely that it may be an ad involving bracelets. Therefore, it is essential to extract information from *keywords* such as "Spark" and "Wrist", and at the same time ignore irrelevant words such as "Your" and "Forever". In the rest of this subsection, we seek to improve the encoding of an ad's text that is guided by its keywords.

[1] The brand name is masked to preserve anonymity.

Keyword extraction. First, we perform *supervised* keyword extraction by training a binary softmax classifier to determine whether each word in the ad's text is a keyword or not based on its hidden representation $\mathbf{h}_i$. Each training instance therefore corresponds to a word. We set its class label to '1' if its POS tag corresponds to a noun, a verb, or an adjective, and '0' otherwise. Effectively, we consider a word to be a keyword if it belongs to one of these three broad syntactic categories. Nevertheless, it does not imply that the model will learn all and only the words belonging to these categories as keywords. Recall that keyword extraction is trained (as an auxiliary task) jointly with ad representation learning in a multi-task setting, so the model's decision on which words will be keywords is in part influenced by the ad representation task.

Text encoding. Next, we use the extracted keyphrases to create a representation of the text portion of the ad that retains its most important information via a Keyword-guided Selective Gate (KSG) mechanism. First, we combine the representations of the keywords as follows:

$$\mathbf{s} = \lambda_1 \mathbf{h}_1 \oplus \lambda_2 \mathbf{h}_2 \oplus \ldots \lambda_n \mathbf{h}_n \tag{1}$$

where λ_i is the keyword extraction model's prediction of whether word i is a keyword. Specifically, λ_i is 1 if i is a keyword and 0 otherwise. Then we utilize $\mathbf{s}$ to generate a selective signal that measures how much semantics of each word in the text should contribute to its context representation:

$$keyGate_i = \sigma(\mathbf{W}_w \mathbf{h}_i + \mathbf{W}_c \mathbf{s}) \tag{2}$$

Based on $keyGate_i$, we filter information of $\mathbf{h}_i$:

$$\mathbf{h}_i{}' = keyGate_i \odot \mathbf{h}_i \tag{3}$$

where $\odot$ represents element-wise multiplication. Then we can generate the selective context representation of an ad's text as follows:

$$\overline{\mathbf{h}} = \mathbf{h}_1{}' \oplus \mathbf{h}_2{}' \oplus \ldots \mathbf{h}_n{}' \tag{4}$$

Using the keywords-guided selective gate mechanism, keywords will contribute more semantics to the context representation. For example, the words "Spark" and "Wrist" are more valuable than "Your" and "Forever" in the text shown in Figure 2.

3.3 Semantic-enhanced Region Extraction

To extract image region features, existing works resort to pre-trained object detection models and keep the top k extracted region features based on the confidence scores that measure how likely the object belongs to the fixed set of categories. However, ad images usually contain a large portion of irrelevant objects that could mislead our ad similarity matching procedure. Without considering the internal context, it is highly likely that the bracelet in the image in Figure 2 will be ignored as it only occupies only a small number of pixels.

In light of this weakness, we propose to improve image region extraction in this subsection by considering the interaction between ad texts and images. Specifically, we use the semantics extracted from the texts to re-weigh image regions so that the object regions related to the promoted products will be given larger weights. For example, the bracelet in the image in Figure 2 will have larger weights based on the semantics of "Spark" and "Wrist".

We implement this idea as follows. First, we extract the top k_1 candidate image regions with the highest confidence scores generated by YOLOv3, and feed the extracted region features to a single-layer feed-forward network (FFN) as follows:

$$\mathbf{v}_i = \mathrm{FFN}_r(\mathbf{r}_i) \tag{5}$$

where $\mathbf{v}_i \in \mathbf{R}^{d_{model}}$. To re-weigh regions, we propose a Semantic Clue Attention (SCA) mechanism, where we use the selective context representation derived in the previous subsection as supervisory signals to give each region a semantic relevance score. Specifically, we attend to the top k_1 regions $\{\mathbf{v}_1, ..., \mathbf{v}_{k_1}\}$ with respect to $\overline{\mathbf{h}}$:

$$\alpha_j = \sigma(\mathbf{W}_v \mathbf{v}_j + \mathbf{W}_o \overline{\mathbf{h}}), j \in [1, k_1] \tag{6}$$

where α_j is the "semantic relevance" score between the j-th region and the key information of the text. Using the relevance value α_j, we re-sort the initial top k_1 regions provided by YOLOv3 and take the top k_2 region features as the final fine-grained visual features to represent an image.

3.4 Dual-path Fusion Attention Layer

Next we fuse the information extracted from an ad's text and image. The input modalities may contain non-informative or misleading information. To address this issue as well as fuse modalities, we propose a Dual-path Fusion Attention (DFA) to generate an ad's multimodal representation. First, we use each modality to refine the features of the other modality based on the confidence of its own inputs. Specifically, the features of one modality are attended to the other as follows:

$$\beta_{ij} = \mathbf{h}_i'^{T} \mathbf{W}_f \mathbf{v}_j, i \in [1, n], j \in [1, k_2] \tag{7}$$

To refine the features from text, the semantic features are calculated by weighted sum as follows:

$$\widetilde{\mathbf{v}}_j = \sum_{i=1}^{n} \frac{exp(\beta_{ij})}{\sum_{b=1}^{n} exp(\beta_{bj})} \mathbf{h}_i' \tag{8}$$

where $\widetilde{\mathbf{v}}_j$ is the refined feature based on $\mathbf{v}_j$. Conversely, the refined image features are computed as:

$$\widetilde{\mathbf{h}}_i = \sum_{j=1}^{k_2} \frac{exp(\beta_{ij})}{\sum_{b=1}^{k_2} exp(\beta_{ib})} \mathbf{v}_j \tag{9}$$

where $\widetilde{\mathbf{h}}_i$ is the refined features based on $\mathbf{h}_i$. Then the refined and original features are fed to a fully-connected layer combined with max-pooling to decide which information should be passed.

$$\widehat{\mathbf{h}}_i = \mathrm{FFN_h}([\mathbf{h}_i'; \widetilde{\mathbf{h}}_i]), \tag{10}$$

$$\widehat{\mathbf{v}}_j = \mathrm{FFN_v}([\mathbf{v}_j; \widetilde{\mathbf{v}}_j]) \tag{11}$$

where $[;]$ denotes the concatenation operation. $\widehat{\mathbf{h}}_i$ and $\widehat{\mathbf{v}}_j$ are fine-grained fusion features. Finally, max-pooling is applied to retain globally useful information: $\breve{\mathbf{h}} = \max(\widehat{\mathbf{h}}_1, \widehat{\mathbf{h}}_2, \ldots \widehat{\mathbf{h}}_n)$ and $\breve{\mathbf{v}} = \max(\widehat{\mathbf{v}}_1, \widehat{\mathbf{v}}_2, \ldots \widehat{\mathbf{v}}_{k_2})$, which are then concatenated to generate the multimodal representation of an ad:

$$\mathbf{f} = \mathrm{FFN}_f([\breve{\mathbf{h}}; \breve{\mathbf{v}}]), \tag{12}$$

By construction, $\mathbf{f}$ contains fine-grained multimodal information.

3.5 Training

To learn ad representations, we utilize triplet loss (Schroff et al., 2015) as the loss function. Given an ad t and its embedding $\mathbf{f}_t$, we constrain it through $\|\mathbf{f}_t\|_2 = 1$ and ensure that each ad $\mathbf{f}_t$ is closer to all other ads $\mathbf{f}_g$ promoting the same product category (positive) than it is to any ad $\mathbf{f}_u$ promoting different product categories (negative). The total loss is calculated as follows:

$$\mathbb{L} = \sum_{\forall(\mathbf{f}_t, \mathbf{f}_g, \mathbf{f}_u) \in \mathcal{T}} \left[\gamma - \|\mathbf{f}_t - \mathbf{f}_g\|_2^2 + \|\mathbf{f}_t - \mathbf{f}_u\|_2^2\right]_+ \tag{13}$$

Dataset	Train	Validation	Test
MP	9413/350	1000/180	3613/390
MS-COCO	16625/60	2375/60	6211/80

Table 1: Dataset statistics of samples/categories. There are 40 and 20 categories in the MP and MS-COCO test sets that are not seen in training and validation.

where γ is a hyper parameter and $\mathcal{T}$ is the set of all possible triplets. Given all labeled ads, we need to calculate all possible triplets, which is computationally expensive. To ensure fast coverage, we choose to learn from the hardest triplets only. Specifically, we take an online strategy to generate triplets from a mini-batch. For each ad in a mini-batch, we obtain the hardest positive sample, $\hat{g} = \arg\max_{g \neq t} \|\mathbf{f}_t - \mathbf{f}_g\|_2^2$, and the hardest negative sample, $\hat{u} = \arg\min_u \|\mathbf{f}_t - \mathbf{f}_u\|_2^2$. The final loss is calculated as:

$$\mathbb{L} = \sum_{t=1}^{l} \left[\gamma - \|\mathbf{f}_t - \mathbf{f}_{\hat{g}}\|_2^2 + \|\mathbf{f}_t - \mathbf{f}_{\hat{u}}\|_2^2\right]_+ \tag{14}$$

where l is the total number of training samples.

Recall that our model jointly learns keyword extraction and ad representations. To learn keyword extraction, we leverage the cross-entropy loss. The overall loss is the weighted sum of the two tasks.

4 Evaluation

The goals of our evaluation are two-fold. First, we evaluate ION's effectiveness in retrieving ads. Second, we evaluate its ability to improve user targeting in real-world application scenarios.

4.1 Datasets

We employ two datasets for evaluation.

MP is a proprietary Chinese ad dataset owned by Tencent. Each ad comprises text and an image. The portion of the dataset that we use contains 14026 ads with 390 product categories. A portion of the test set is composed of ads belonging to 40 product categories that do not appear in the training or validation sets. This will allow us to evaluate our model's ability to generalize to new product categories.

MS-COCO (Chen et al., 2015) is a large public text-image matching dataset. Though it is not an ad dataset, we use it because (1) there is currently no public dataset for retrieving ads of the same product category, (2) each sample has text, image and object categories, which is similar to ad samples, and (3) existing multimodal datasets collected for

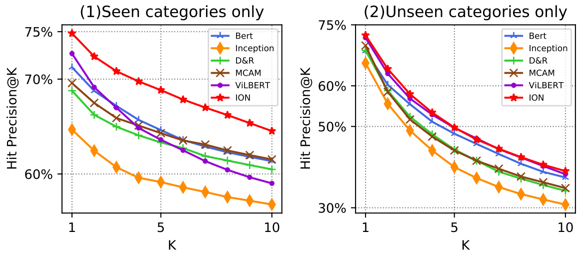


Figure 3: Results on MP.

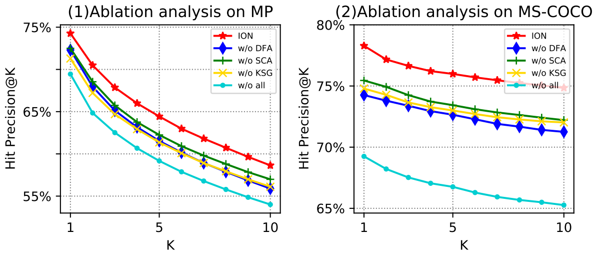


Figure 5: Ablation results on MP and MS-COCO.

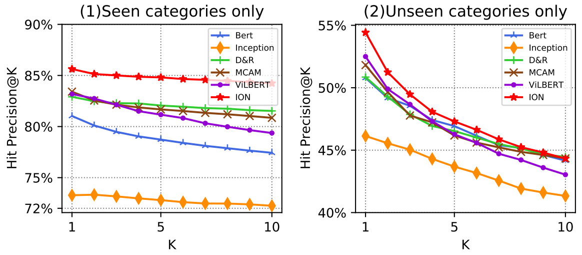


Figure 4: Results on MS-COCO.

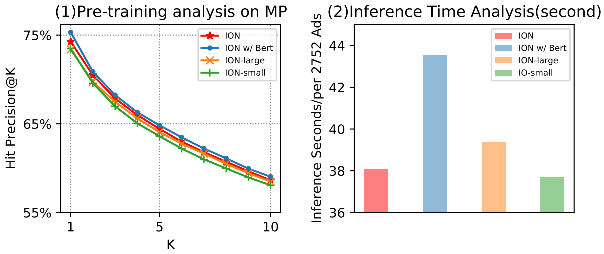


Figure 6: Pre-training and inference time analysis on MP.

specific tasks, such as visual question answering, multimodal sarcasm, are not consistent with our experimental setup. We only retain samples that belong to one object category. There are 25211 samples with 80 labels. As in MP, a portion of our test set in MS-COCO is composed of ads belonging to 20 categories that do not appear in the training or validation sets. Statistics on these datasets are shown in Table 1.

4.2 Implementation Details

We exploit jieba to segment Chinese ad text in MP. The input image size is 416×416×3 and d_{model} is 128. Other parameters are tuned using grid search. The Transformer we use contains 4 multi-head layers and the head number in each layer is 4. For region detection, we use pretrained YOLOv3 and take outputs of the last layer as region features. k_1 and k_2 are 20 and 10. For training, γ in the loss function is 0.2 for MP and 0.3 for MS-COCO. In our model's loss function, we set the weight of the ad representation learning task to 1 and that of the keyword extraction task to 0.05. The Adam optimizer with learning rate e^{-3} is used. All models are trained on Tesla V100 with 32GB memory for 30 epochs with batch size 32, and the best epoch based on the validation loss is selected for testing. We use each sample s in the test set to query all other samples in the test set to obtain the top k ads that are most similar to s, where the distance between two ads is the Euclidean distance between their ad representations as learned by our model.

In real application scenarios, it is useless to recall all possible candidates and the top results are more than adequate, so we employ Hit P@k (precision within the top k results) as our evaluation metric.

4.3 Baseline Systems

We compare ION with five baselines that include text-only, image-only and multimodal methods. The text baseline is BERT (Devlin et al., 2019), which has achieved prominent performance in many language processing tasks. We weigh-sum the last layer of BERT's output as the text representation. As the image baseline, we employ the most commonly used Inceptionv3 (Szegedy et al., 2016), which is pretrained on the ImageNet dataset. As multimodal baselines, we employ D&R (Xu et al., 2020) and MCAM (Abavisani et al., 2020), which are the state-of-the-art multimodal networks of their respective tasks. Furthermore, we compare with the multimodal pre-trained model ViLBERT (Lu et al., 2019), which has achieved impressive performance in numerous text-image tasks. The baselines' parameter settings are the same as those reported in their respective papers.

4.4 Results and Discussion

Figure 3(1) and Figure 4(1) show the results of ION and our baselines on the portion of the test sets in MP and MS-COCO where the product categories are seen during training. We present these results in the form of a graph where HIT P@k is plotted against k (the number of ads retrieved). As can

k_1	k_2	P@1	P@5	P@10
20	1	0.7197	0.6240	0.5712
	5	0.7349	0.6336	0.5788
	10	**0.7427**	**0.6443**	**0.5864**
	15	0.7421	0.6431	0.5845
	20	0.7393	0.6393	0.5823
10	10	0.7396	0.6436	0.5846
30		0.7384	0.6356	0.5791

Table 2: ION performance with different k_1, k_2 pairs.

be seen, ION achieves the best results, obtaining a 74.8% HIT P@1 on MP and a 85.6% on MS-COCO. The image-only baseline performs worse than the text-only baseline because images contain a lot of background noise. Our multimodal baselines, R&D and MCAM, outperform the image-only baseline by a large margin. These results demonstrate the necessity of considering both texts and images. Nevertheless, the multimodal baselines extract coarse-grained features from images without considering the local correlations between modalities and fail to curb the bad influence of unrelated pixels. As a result, they perform worse than ION. It is worth noting that BERT and ViLBERT have benefitted from large corpora for pre-training and thus outperform both multimodal baselines.

Next, to verify ION's generalization capability, we compare ION with our baselines on the portion of the test sets in MP and MS-COCO where the product categories are not seen during training. As shown in Figure 3(2) and Figure 4(2), ION outperforms all baselines, which demonstrates the better generalization of our model.

4.5 Additional Experiments with ION

Ablation experiments. We perform three ablation experiments to verify the effectiveness of each component in ION. First, we ablate Keyword-guided Selective Gate (KSG) (Section 3.2) simply by taking the representation from Transformer as the word representation. We denote this as *w/o KSG*. Next, to ablate Semantic Clue Attention (SCA) (Section 3.3), we retain the top k_2 regions based on YOLOv3 scores instead of reranking the detected regions. We denote this as *w/o SCA*. Finally, we ablate Dual-path Fusion (DFA) (Section 3.4) by replacing it with global concatenation fusion. Specifically, we apply max-pooling over the text representations and the image representations, and then concatenate them to create the fusion representation. We denote this as *w/o DFA*. Moreover, we have an experiment where we ablate all three mechanisms. We denote this as *w/o all*. Ablation results are shown in Figure 5. As can be seen, removing any of them negatively impacts model performance.

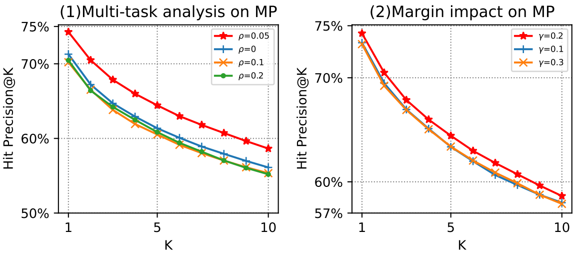
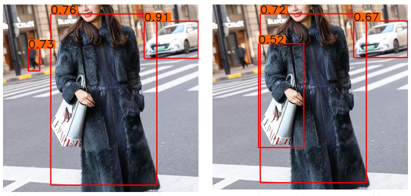


Figure 7: The multi-task learning and γ analysis.



Custom-tailored Overcoat, this Winter no Cold

Figure 8: Example of image region re-weighting.

Effect of encoder size. How will the results differ if a larger/smaller Transformer is used? As shown in Figure 6, ION with a larger Transformer (8 layers and 8 heads) or a smaller Transformer (1 layer and 1 head) both exhibit a deterioration in performance. The reason is that Large Transformer needs more data to learn better, while Small Transformer may not be able to encode everything needed to perform well. We also replace Transformer with BERT. While pre-training BERT optimizes ION, it considerably increases inference time and leads to low efficiency. To achieve high precision and efficiency, a smaller Transformer encoder would therefore suffice. Our model can complete the inference of 1 million ads in 3.8 hours using a single-machine system, which meets the requirement of real scenarios.

Impact of regions. We evaluate the ION performance with different settings of k_1, k_2 pair in detecting image regions on MP. We vary k_1 within {10, 20, 30} by fixing k_2 to 10 and k_2 within {1, 5, 10, 15, 20} by fixing k_1 to 20. As shown in Table 2, ION works best with k_1=20 and k_2=10. The small k_2 results in insufficient visual features, and the large k_2 shows weakness on the grounds of background noise. k_1 has a similar impact on ION performance.

Impact of multi-task learning. We analyze how the two tasks affect the learning process by varying the keyword extraction's weight in the total loss. We set the keyword extraction weight ρ to 0, 0.05, 0.1, and 0.2. As shown in Figure 7, ION performs worse as ρ increases since a larger ρ produces more bias to the auxiliary task and results in insufficient training for the main task. Considering the ρ=0 results, we can see that learning to extract keyword improves ION.

Impact of γ. We also analyze how the triplet loss margin parameter γ impacts ION. As shown in Figure 7, ION achieves the best result with γ=0.2.

4.6 Qualitative Analyses

An example. We begin this subsection by illustrating how ION works via an example. Specifically, we visualize the region detection performance with and without SCA in Figure 8, which shows the top 3 regions and their scores before (left) and after (right) semantic based re-weighting. It is clear that using text as clues gives larger scores to product related regions and decreases those corresponding to background noise.

Error analysis. To gain insights into why ION offers superior performance to ViLBERT, we randomly select 100 samples from the test set for which the similar ads are recalled incorrectly by ViLBERT and correctly by ION and analyze these samples. We found that ViLBERT typically extracts regions that have obvious object features, specifically objects that take up a major portion of an image, but the extracted regions are unrelated to the ad product. In contrast, ION was able to focus on product related regions. For example, ViLBERT incorrectly recalls a sofa ad for a jeans ad. The reason is that a model who is dressed in jeans is sitting on a sofa in the jeans ad, and ViLBERT treats the jeans ad as a sofa ad because the sofa has more obvious object features. In contrast, guided by textual information, ION successfully recognizes jeans. Another example involves a watch ad. ViLBERT incorrectly recalls a coat ad because the models wearing the coat/watch occupy a large portion of the images and are similar to each other, whereas ION avoids this problem by paying attention to the fine-grained region occupied by the product.

It is interesting to note that not all test samples that are correctly classified by ViLBERT are also correctly classified by ION. To better understand how ViLBERT is better than ION, we randomly selected another 500 samples in the test set for which ViLBERT was correct but ION was wrong. We found that ION has a bias towards image shape features, which means that ION prefers to recall ads with similar product shapes. As mentioned before, ION focuses on product-related regions. If the shapes of two products are similar, ION would assume the corresponding ads are similar. For example, ION incorrectly recalls pen ads for lipstick ads because the shape of pens and that of lipsticks are both cylindrical. In contrast, ViLBERT does not have this bias.

4.7 Experiments on User Targeting

To verify ION's ability to improve user targeting models (i.e., whether the idea of augmenting positive instances using users clicking on ads with the same category works in real scenarios), we conduct offline and simulation user targeting experiments.

The offline experiment. In this experiment, we assemble a dataset for evaluating ION as follows. We select from a database an initial ad package and collect the clicking users over a certain period of time P. These clicking users constitute the positive instances in the dataset. To get the negative instances, we randomly sample from I, the set of impression users (users who have seen the initial ad package). To avoid a skewed class distribution, we maintain a positive to negative ratio of 1:3, which is the standard in the ad industry. We then reserve 10% of the users in this dataset for testing (and call this test set T), and use the remaining 90% to train a user targeting model, which we call $M_{initial}$. Next, to evaluate how effective ION is, we use ION to find the 10 ads that most likely belong to the same category as the initial ad package and use the clicking users of these 10 ads to augment the positive training set used to train $M_{initial}$. Given this augmented set of positive training instances, we also augment the negative instances by randomly sampling from I until the desired ratio of 1:3 is reached. Finally, we use this augmented set of positive and negative training instances to train a user targeting model, which we will denote as $M_{expanded}$. We evaluate $M_{initial}$ and $M_{expanded}$ on T. Figure 9 depicts this experimental procedure.

The simulation experiment. In the simulation experiment, the test set is constructed by collecting user clicks in real world. Specifically, we collect over a certain period of time P', which would be

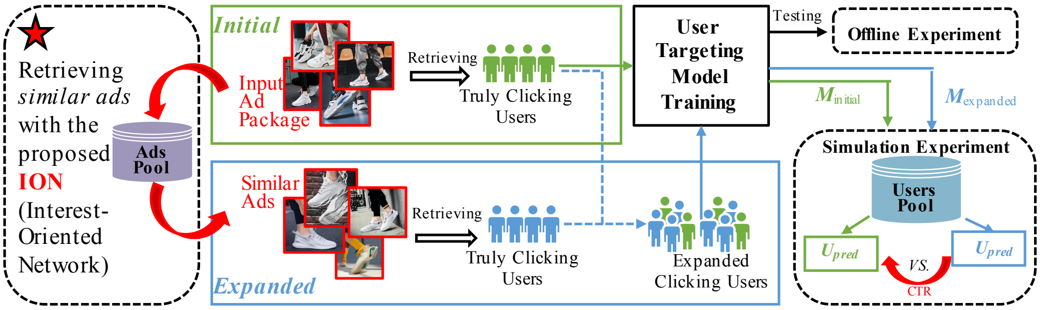


Figure 9: Procedure for conducting the user targeting experiments.

sometime after time period P, the set of impression users I' in P' who click on the initial ad package during P' and denote the resulting set of users as U_{click}. Note that U_{click} is the set of ground-truth clicking users. We then use $M_{initial}$ to retrieve the targeted users of the initial ad package from I' and denote the resulting set of users as U_{pred}. We compute the Clickthrough Rate (CTR) as $U_{pred} \cap U_{click}$ divided by U_{pred}. Note that a larger CTR value implies that a user targeting model is better at recalling potential clicking users. We similarly user $M_{expanded}$ to retrieve the targeted users and compute its CTR.

User targeting model. We employ XGBoost (Chen and Guestrin, 2016) to train our user targeting model. XGBoost provides a regularizing gradient boosting framework that is commonly used to train models to predict click-through rates. The inputs are the features of an user and the output is 0/1 denoting click/non-click the ad package. In our experiments, we use 57 user features, such as property status, geographic location, and education level, which are encoded as one-hot vectors. We employ CART as the base classifier. The max depth of CART is set to 6, the learning rate is 0.1, and the number of gradient boosted trees is 550.

Dataset. As our dataset, we collect eight ad packages with low click-through rates (i.e., rates between 0.2% and 0.9%). Before augmentation by ION, there are on average 3305 positive users (i.e., users who clicked ad packages) per package. After augmentation, there are on average 59037 positive users per package.

Results. Table 3 shows the average performance obtained by repeating the offline and simulation experiment 8 times with 8 ad packages. Compared with $M_{initial}$, all $M_{expanded}$ yield increases in the CTR value, thus demonstrating that augmenting positive instances with user clicking data from the same category works in real scenarios. Importantly, ION achieves a greater degree of improvement on AUC and CTR than the baselines do, which should not be surprising as it is more accurate in determining which ads belong to the same category.

	Offline	Simulation
Model	AUC	$CTR(‰)$
Initial	0.75498	7.355
Bert	0.79237	11.01
Inception	0.76569	8.72
D&R	0.78943	10.713
MCAM	0.79498	11.941
ViLBERT	0.80583	12.191
ION	0.81413	13.177

Table 3: Results of offline and simulation experiments.

5 Conclusions

We proposed to alleviate the insufficient positive instance problem associated with the training of user targeting models by retrieving ads for the same product category as that of the ad package under consideration via a novel bimodal model, ION, and then using their clicking users for data augmentation. Results on two datasets showed that ION can effectively retrieve ads belonging to the same category and improve a user targeting model.

Acknowledgments

We thank the three anonymous reviewers for their comments on an earlier draft of this paper. This work was supported in part by the National Natural Science Foundation of China (No. 61802167), the US National Science Foundation (Grant IIS-1528037) and Tencent. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of the funding agencies.

References

Mahdi Abavisani, Liwei Wu, Shengli Hu, Joel R. Tetreault, and Alejandro Jaimes. 2020. Multimodal categorization of crisis events in social media. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14667–14677. IEEE.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1657–1668.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.

Guillem Collell, Ted Zhang, and Marie-Francine Moens. 2017. Imagined visual representations as multimodal embeddings. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4378–4384. AAAI Press.

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 191–198.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Zhibin Hu, Yongsheng Luo, Jiong Lin, Yan Yan, and Jian Chen. 2019. Multi-level visual-semantic alignments with relation-wise dual attention network for image and text matching. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 789–795. ijcai.org.

Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. 2018. Learning semantic concepts and order for image and sentence matching. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6163–6171. IEEE Computer Society.

Andrej Karpathy and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, pages 212–228.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5070–5081.

David G. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23.

David Novotný, Diane Larlus, and Andrea Vedaldi. 2017. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2867–2876.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2249–2255.

Jinwei Qi, Yuxin Peng, and Yuxin Yuan. 2018. Cross-media multi-level alignment with relation attention network. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 892–898. ijcai.org.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823.

Botian Shi, Lei Ji, Pan Lu, Zhendong Niu, and Nan Duan. 2019. Knowledge aware semantic concept expansion for image-text matching. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5182–5189.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566.

Ming Tan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2016. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure-preserving image-text embeddings. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5005–5013.

Qianqian Wang, Xiaowei Zhou, and Kostas Daniilidis. 2018. Multi-image semantic matching by mining consistent features. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 685–694.

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017a. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17, Halifax, NS, Canada, August 13 - 17, 2017*, pages 12:1–12:7.

Yaxiong Wang, Hao Yang, Xueming Qian, Lin Ma, Jing Lu, Biao Li, and Xin Fan. 2019. Position focused attention network for image-text matching. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3792–3798.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017b. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4144–4150.

Nan Xu, Zhixiong Zeng, and Wenji Mao. 2020. Reasoning with multimodal sarcastic tweets via modeling cross-modality contrast and semantic association. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3777–3786. Association for Computational Linguistics.

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4699–4709.

Zhedong Zheng, Liang Zheng, Michael Garrett, Yi Yang, Mingliang Xu, and Yi-Dong Shen. 2020. Dual-path convolutional image-text embeddings with instance loss. *ACM Trans. Multim. Comput. Commun. Appl.*, 16(2):51:1–51:23.

Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1059–1068.