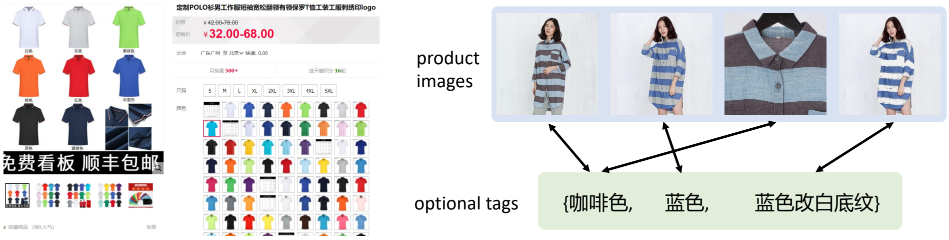
EMIT: Embedding Matching for Image-Text pairing

Helin Xu*



定制POLO衫男工作服短袖宽松翻领有领保罗T恤工装工服刺绣印logo

价格 ¥~~42.00-78.00~~

促销价 ¥32.00-68.00

运费 广东广州 至 北京 快递: 0.00

月销量 500+ | 送天猫积分 16起

尺码 S M L XL 2XL 3XL 4XL 5XL

颜色

收藏商品 (881人气) 举报

Figure 1. Overview of image-text pairing task. Our method EMIT pairs images and text tags through embedding matching.

Abstract

In this work, we tackle the problem of image-text pairing, i.e. pairing images of various kinds of clothes and their corresponding text tags scrapped from Taobao. We propose to extract feature embeddings from both the images and the text tags and then match the embeddings of images and texts in the feature space. Our Embedding Matching for Image-Text pairing method achieves an impressive improvement compared to TAs' baseline methods. Moreover, to better leverage the constraint of images covering all optional tags, we design a heuristic-based bagging process that further boosts our performance. Codes are available at https://www.github.com/HelinXu/EMIT.

1. Introduction

Taobao (https://www.taobao.com/), the biggest online store in China, hosts billions of products on sale, among which clothes are a typical case. There are a million kinds of cloth (*i.e.* products) on Taobao, each of which comes with a couple of styles (*i.e.* options). We hereby wish to develop a method that streamlines the process of manually labeling all the images, and thus comes the task of image-text pairing.

We propose a novel approach that pairs images and text tags through embedding matching. (a) We adopt an image feature extractor for image feature embedding and compute word embedding for each piece of text tag. (b) Cross entropy is used to match the image and text embedding together during training, and for selecting the best-matching image-text pairs during inference. (c) Heuristic-based bagging is designed to overcome the drawback of the embedding matching method. Our main contributions are as follows.

1. We propose to decompose the text tags into semantic labels, and propose an algorithm, EMIT, of embedding matching.
2. At the end, we further design a heuristic-based bagging method to boost accuracy.
3. Our method significantly outperforms existing baselines on Taobao Dataset, and we provide ablation studies to validate the effectiveness of our method.

2. Task Formulation

In this section, we further formulate the task.

Image-text pairing aims to match all the images from the same product to all the text tags. In this paper, we use **product** to refer to a single product on sale on Taobao, which often comes in several styles or options with typically several images under each style or option. **Optional tags** refers to the text-based tags that tell different styles from each other within the same product. For example, a certain kind of skirt is a product, and this product comes in three different styles, pink, white, and black. The text labels pink, white, black are the optional tags for this certain product.

For a certain product, we have n images in total

$$\mathcal{I} = \{I_1, I_2, ..., I_n\}$$

*: Department of Automation, Tsinghua University, ID 2019011430, xuhelin1911@gmail.com

'牛油果绿预售15天', '【杏色】单件西装', '蓝色T恤+蓝色长裤', '烟雾粉181', '白色上衣+黑色套装裙子', '2801XCY紫色', '粉色上衣黑色阔腿裤套装', '军绿风衣', '丈青色上衣+卡其色裤子', '灰色套装', '黑白格923', '黑红', '323_灰色', '蓝色T恤', '黑白条纹', '紫红色连衣裙_', '松雪红', 'F5J7-红色', '单件灰白格中袖西装', '嫩黄色3082', '9665#红色', '[E]红色', '豆绿色', '粉橘色', '0505_驼色', '变色绿', '浅沙', '人字纹驼色', '粉色裤子', '杏白色开衫', 'BZ1609焦糖色', '218红色', '黑色-019', '暗紫', '粉色鹿角卫衣'

Figure 2. Messy optional tags.

with m different optional tags

$$\mathcal{T} = \{T_1, T_2, ..., T_m\}$$

For each product, we output the image-text pairs for all images in the product,

$$(I_1, T_{I_1}), (I_2, T_{I_2}), ..., (I_n, T_{I_n})$$

where tags are selected from its optional tags,

$$\forall i \in \{1, 2, \ldots, n\}, I_i \in \mathcal{I}, T_{I_i} \in \mathcal{T}$$

3. Task Analysis

With careful inspection, we find this task challenging especially due to the messy optional tags as well as corner cases. In this section, we provide a detailed analysis of the challenges in this task, followed by our novel designs tailored for all of them.

First and foremost, the dataset is raw in terms of optional tags. After enumerating the dataset, we find optional tags messy. There are over 5,000 kinds of unique optional tags among all products in the dataset. Treating the task as a naive classification task with optional tags as labels is not ideal. A selection of optional tags is shown in the Figure 2.

There are these challenges that lie in the dealing with optional tags:

1. The inclusion of meaningless characters within the optional tags (e.g. ’BZ1609’);

2. The inclusion of unrelated semantic information, such as cloth styles (e.g. ’T-shirt’, ’Dotted’), sales-related information (e.g. ’quick delivery’);

3. The cases where there is more than one single item in one product (e.g. when a product includes a full suite of cloth, such as ’white T-shirt + black pants’);

4. The internal complexity of text-based color information. Color-related information can be further decomposed into three components: (1) The lighting of a certain color (e.g. dark, light, etc.); (2) The texture (e.g. dotted, stripped, etc.); (3) The color itself (e.g. blue, green, etc.).

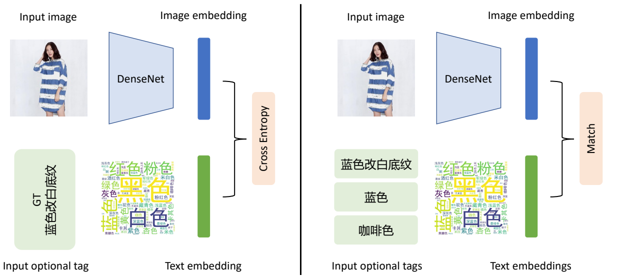


Figure 3. Method overview. (1) Left: during training, a DenseNet[1]-based feature extractor is trained on the text embeddings in a multi-label classification fashion, to match the image embedding to the text embedding. (2) Right: During inference, we extract the image embedding from the testing image and all text embeddings from the optional tags, and we match them by selecting the text embedding that has the smallest cross-entropy with the image embedding.

We hereby propose to first decompose the optional tags into separate words using Jieba (https://github.com/fxsjy/jieba), to treat all kinds of information in a decomposed manner, so that it’s possible for us to filter out the unrelated semantic information while keeping the information that matters. In this way, we obtain the following advantages:

1. We get the number of labels under control;

2. We keep the labels that matter by sorting the labels;

3. Our design of encoding cover several modalities of an image (lighting, texture, color, etc.) so that we made learning explainable by guiding the network to understand the separate concepts of different modalities.

4. Method

4.1. Method Overview

We propose a novel approach that pairs images and text tags through embedding matching, Embedding Matching for Image-Text pairing (EMIT), as shown in Figure 3.

We first decompose the optional tags into separate words using Jieba and encode the optional tags and extract the text embeddings, which is a 0,1 vector. Then, a DenseNet[1]-based feature extractor is trained on the text embeddings in a multi-label classification fashion, to match the image embedding to the text embedding. This is where the network learns about the separate semantic labels on this image. During inference, we extract the image embedding from the testing image and all text embeddings from the optional tags, and we match them by selecting the text embedding that has the smallest cross-entropy with the image embedding.

Last, we further propose a heuristic-based bagging method that takes into account the constraint that all images from one product should make a full cover over the optional tags of this product. What we do is that we train several different networks using a different selection of semantic labels, data dropouts, etc., and obtain several models for inference. For each product, we reject the predictions from the models where this constraint was not met while keeping the predictions that meet this constraint. We show that by considering this constraint, we further boost the performance of our method.

Data processing. The data processing procedure is crucial in our method.

1. Jieba word cutting. We decompose the messy optional tags into words (semantic labels) so that semantic information is decomposed.

2. Label sorting and filtering. We sort the semantic labels on the full dataset. Semantic labels that appear less than 5 times over the full dataset are considered noise and thus filtered. The dominating semantic labels are colors. However, we also get labels covering other semantic information such as 'T-shirt', 'suite', 'dark', etc.

3. Multi-label assigning. One image can have multiple labels, so that network will understand and disentangle different semantic information. For example, 'grey white dotted T-shirt' gets grey, white, dot, T-shirt.

This process is rather fast. In practice, we do the Jieba word cutting, sorting, and filtering beforehand, save the top 64 dominating semantic labels to label set $\mathcal{S}$, and generate the vector-like text-embedding on the fly.

Text embedding. We retrieve text embedding F_T by looking into the $\mathcal{S}$, and assigning 1 to $F_t[i]$ if $\mathcal{S}[i]$ is a substring in the optional tag.

Image embedding. We reshape all images to 224*224 and use a DenseNet[1]-like neural network as the backbone. In the last layer, we add a fully connected layer with 64 output channels indicating 64 different semantic labels that we are learning, and we would like the image embedding F_I to match the text embedding on each corresponding channel.

Loss function. We use cross-entropy loss.

Optimizer. We use Adam optimizer with a learning rate of 1e-3.

5. Experiment

We train our method on the medium-sized dataset with batch size = 64. We use LR = 1e-3 and our method can converge when trained on a single GeForce 3090 GPU for about 30 minutes.

Experiment results. With no bells and whistles, our method, EMIT, outperforms the baseline methods by a large margin, with results shown in Table 1.

Ablation studies. We further do ablation experiments to validate the effectiveness of our design of multi-label text-embedding and our design of heuristic-based bagging. Results are shown in Table 2.

Method	Acc	EM
TA_000	0.8839	0.6243
TA_001	0.8673	0.6155
EMIT(Ours)	**0.9477**	**0.8040**

Table 1. Experimental results.

Experiment	Semantic label	Bagging	Acc	EM
1	Manual design	No	0.8429	0.5519
2	Text embedding	No	0.9231	0.7526
3(Ours)	Text embedding	Yes	**0.9477**	**0.8040**

Table 2. Ablation studies.

(1) Experiment 1 is done as described in the assignment. We adopt the same DenseNet-like network structure as used in our method and only switched to manually designed color labels. In particular, we keep all the color labels that we adopt in our EMIT version and do a single-class classification on the dataset. Experiments 1 and 2 show that keeping not only the color labels but also other information like 'T-shirt', 'suite', 'dark', etc. and treating the problem as a multi-label classification benefits the performance.

(2) We run experiment 2 several times with different selection of semantic labels, data dropouts, etc. The top-performing model reaches the accuracy in the table. However, from experiments 2 and 3 we can see that putting each model together by our heuristic-based bagging method further improves the accuracy by 2 percent and the EM by 5 percent.

6. Conclusion

In this paper, we tackle the problem of image-text pairing on Taobao Dataset. To achieve this goal, we propose to decompose the text tags into semantic labels, and propose an algorithm of embedding matching. At the end, we further design a bagging method to boost accuracy. Our method significantly outperforms existing baselines.

References

[1] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2, 3