



PhotoChat

摘要

我们提出了一个新的人与人对话 Photochat，一个反映在线照片分享行为的数据集，包含 12000 个对话。

提出两个任务：

1. 图片分享意图预测任务，预测是否分享照片
2. 照片检索任务，根据对话上下文检索出最相关的照片

introduction

照片分享的情况变多了，需要一个智能分享照片的代理。

照片分享任务的重点是根据对文本的理解选择图片，而之前的工作更多是根据对图片的理解生成文字：

1. 对话往往不会明确提到图片中的主要内容
2. 对话不一定与图像相关，经常包含问候和其他话题的闲聊；与此相反的是，在传统的图像-文本任务中，正确的文本被设计为与图像高度相关，并且很少有分散注意力的内容。

为了解决这些问题，本文创建了 PhotoChat——一个人-人对话的数据集，其中在对话过程中，一个人向另一个人分享照片，这是第一个捕捉照片分享活动的数据集。

数据集来源：从 OpenImageV4 中作为分享的照片，并众包生成 12286 个对话。

两个任务：

1. 照片分享意图检测
2. 基于对话的图像检索任务
 - a. 我们还提出了一个双编码器模型，利用对象标签来编码图像特征，在所有没有交叉注意力机制的模型中取得了最好的性能。

本文的贡献：

1. 人-人对话数据集
2. 两个新任务去实现照片分享系统
3. 为新任务提出了基线模型

Dataset Creation

来源：收集 Open Image Dataset V4 并且在亚马逊上众包了真实对话。

Image-based Filtering

原始的 OID 数据集太多不像会被分享出来的照片，所以进行了筛选，有关四个主题的照片被普遍分享：人、食物、动物和产品（购物场景）

首先列出了属于四个主题的标签，像交通灯、水龙头、钉子等标签被抛出在外，诸如女孩，面包圈，小鸟等标签则包括在内。这个过程共选择了 89 个对象标签。

然后生成了一个图像池，选择那些包含列表内物体的图像。特别注意的是，对于人类类别的物体，我们增加了另一个标准，它必须是照片的主体。图像被随机的在图像库中选择，以便于在下一步生成对话。

Conversation Generation

众包根据图像产生对话，图像带有一个图像描述，其中有图像中的物体标签列表。图像中包含人类的时候，给其中一个人分配一个随机的名字和关系，帮助众包工作人员参考。他们被告知要想象与他们的朋友对话，任务开始时，只有一个人可以接触到图像，并被指示要推动对话，直到它适合于另一方分享图像。

对话次数五次，分享照片后，他们可以继续聊天，直到他们希望结束对话，并提交对话。

Image&text-based Verification

最后，使用另一组内部专业人群来过滤掉上述步骤中产生的无效对话。如果图片和对话之间的联系在照片分享之前就已经不言而喻（in-evident），或者内容不自然，含有不恰当的词汇，太多错别字，都会被过滤掉。




Good Example	Good Example	Bad Example
		
<p>A: hows it going? B: just got back from vacation!! A: How was vacation? did you have fun? B: It was exciting! I took my grand-daughter to Greece and we saw so many beautiful ruins! A: oh wow! Greece, that's amazing. I bet you got amazing pictures of the ruins B: Yeah, we saw ancient temples and battlefields B: Share the photo A: Wow! that's a great photo. you should post it on Insta too. B: Great idea! Thanks!</p>	<p>A: hey guess what i'm doing now ?? B: What are you up to today? A: i'm preparing a pizza for the first time i include tomatoes,onions and so on B: Wow, you must be daring! Whoever taught you should have been confident on your progress. A: hey..... i'm almost done B: Must be yummy, A: wanna see my preparation? A: Share the photo</p>	<p>A: How are you? B: I'm doing well. I've been watching Netflix because I can't go outside. A: Yeah, same here. Which show? A: And actually, I just found this picture of someone who should be a photographer. B: The office has been my go to. B: Really? Share the photo to me. A: Share the photo B: Whoa! You were totally right A: It's a boy in neon green who I think wants to take photos in academic settings. B: This photo is so cool</p>

Figure 2: Examples of PhotoChat dataset. The first two examples are included in the dataset while the last example is excluded in the verification step. **Share the photo** denotes the photo sharing act.

Dataset Statistics

对话分为四类，人，食物，动物，日常用品

训练集：10086，测试集：1000，交叉验证集：1000

每个对话平均回合数 12.7，轮数 6.3，由于不严格限制要交替发言，所以本文把来自同一方的连续会和合并为同一个

Task Definition

将照片推荐系统问题分解为两个独立的任务：

1. 照片分享意图预测任务：检测用户是否有在下一个回合分享照片的意图
2. 图像检索任务：根据对话上下文检索照片

符号定义

令 $P = \{p_1, p_2, \dots, p_M\}$ 为照片序列，其中 $p_i = (a_i, l_i), i \in [1, M]$ ，由图像 a_i 和一系列对象 l_i 组成。

给定对话 $D = \{t_1, \dots, t_h, p_k, t_{h+1}, \dots, t_N\}$ ，其中两个参与者交替发言， $t_j (j \in [1, N])$ 和 $p_j \in P$ 分别代表第 j 轮的对话和分享的照片。

t_h 是照片分享行为之前的回合，定义说话人信息 $S = \{s_1, s_2, \dots, s_N\}$ ，其中 $j \in [1, N]$ ，可以是 0 或者 1，表示第 j 个回合的说话人。

Photo-sharing intent prediction

意图预测目标是预测在之前的所有回合中， t_j 的照片是否会在下一回合被分享，被定义为一个二元分类任务：

$$\forall j \in [1, h], C(t_{1:j}, s_{1:j}) \in \{0, 1\}$$

其中 C 是意图预测模型，包括话语和说话人的信息作为输入，并输出一个 0 或 1， $j=h$ 时为 1，否则为 0。

使用 F1-score, recall, precision 作为评价指标。

Image retrieval

在相同设置下，图像检索任务的模型 R 被期望能从照片序列 P 中检索出 p_k

$$R(t_{1:h}, s_{1:h}, P) \in [1, M]$$

整个训练集中，候选池 P 通常由一个 batch 内的图像组成；在测试期间， P 包含测试集所有的图像。

我们使用 Recall@K 评估，计算为在前 k 个结果中找到一个正确项目的百分比作为评价指标。具体来说，我们选择 $R@1$ 、 $R@5$ 、 $R@10$ 以及他们的综合 $sum(R@1, 5, 10)$ 来评价

Baselines

Photo-sharing Intent Prediction Model

finetune 了 Bert、ALBert 以及 T5

BERT、ALBERT：串联之前的轮次通过[SEP]，首部加上[CLS]作为模型的输入，使用说话人信息作为 segment id 的输入；[CLS]token 被送入全连接层，128→2；

T5:通过[SEP]串联句子，预加"predict share intent:"作为模型的输入。

loss function：交叉熵

Image Retrieval Model

dual encoder

建立双编码器模型，分别使用 sota 预训练模型编码图像和文字

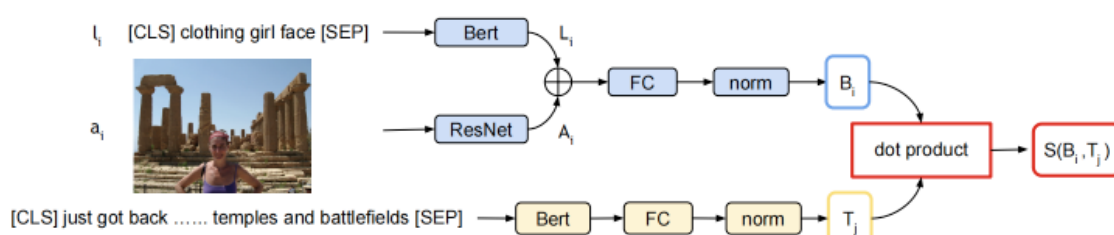


Figure 3: Our dual encoder. The first dialogue in Figure 2 is used as the input example. Image and text are encoded separately to generate their embeddings. The dot product of them is then used to compute the similarity score.

编码图像：对每一个 $p_i = (a_i, l_i)$ 我们首先 resize 图像尺寸为 224*224，然后送进 ResNet 去产生 A_i

编码文字：这里编码的是照片的描述，使用预训练 BERT 模型得到嵌入层的[CLS]的 token： L_i

然后串联 L_i 和 A_i 。

编码对话：使用第二个预训练模型 BERT 编码对话，得到 T_j

分别通过全连接层映射为 H 维后，计算两个向量的点乘相似度，获得相似性分数 $S(B_i, T_j)$

损失函数使用双向 in-batch 交叉熵损失

$$l_{sm}(B_i, T_j) = - \left(S(B_i, T_j) - \log \sum_{\hat{T}_j} e^{S(B_i, \hat{T}_j)} \right) - \left(S(\hat{B}_i, T_j) - \log \sum_{\hat{B}_i} e^{S(\hat{B}_i, T_j)} \right),$$

其中， B_i 和 T_j 是每一个样本的图像嵌入和文本嵌入

本文同样实验了双向 in-batch hinge loss，定义为：

$$l_{sh}(B_i, T_j) = \sum_{\hat{T}_j} \left[\alpha - S(B_i, T_j) + S(B_i, \hat{T}_j) \right]_+ + \sum_{\hat{B}_i} \left[\alpha - S(B_i, T_j) + S(\hat{B}_i, T_j) \right]_+$$

其中 α 是边缘参数，并且 $[x]_+ = \max(x, 0)$ 。在本文中的初步实验中，我们观察到交叉熵损失更好，并且在大多数实验中都是用了交叉熵损失。

VSE++

VSE++ 是一个简单并且有效的双编码器模型。

它编码文字和图像，连接所有先前的句子直到用户分享图片，分别使用 ResNet152 和 GRU 来编码。然后通过线性映射他们到相同的维度。

最后，点乘归一化后的潜在城并且计算相似度分数。

他们创新性地在排名损失函数中使用了最难的否定词，也就是最接近查询的否定词。

$$l_{mh}(B_i, T_j) = \left[\alpha - S(B_i, T_j) + S(B_i, \hat{T}_j^h) \right]_+ + \left[\alpha - S(B_i, T_j) + S(\hat{B}_i^h, T_j) \right]_+$$

其中， $\hat{T}_j^h = \operatorname{argmax} (S(B_i, \hat{T}_j))$ 和 $\hat{B}_i^h = \operatorname{argmax} (S(\hat{B}_i, T_j))$ 是最难的否定词。

$$\hat{T}_j^h = \operatorname{argmax} (S(B_i, \hat{T}_j))$$

SCAN

scan 是一个完整的交叉注意力模型，可以捕捉图像区域和文本标记之间细粒度的相互作用，以推断出图像-文本的相似性。

使用 fasterRCNN 和 ResNet-101 区计算图像区域的嵌入矩阵，使用双向 GRU 来计算文本嵌入。

损失函数与 VSE++ 相同。

虽然他在图像标题任务上比 VSE++ 要好，但是交叉注意力计算的成本比较高，他不能扩展到大规模的图文检索问题中。

BM25

BM25 是一个广泛用于文档检索的概率性检索函数，为了使其适应我们的设置，直接使用每张图片的标签作为文档术语。

照片分享之前，所有的话语被串联起来，作为查询词来检索图片。

Experiments

Setup

照片分享意图预测：BERT，ALBERT，T5 的最大序列长度为 512，选用最高 F1-score 来作为模型选取的办法。

baseline：BERT 的最大序列长度 128，文本嵌入维度 $H=512$ ，边缘参数 $\alpha = 0.2$ 。

训练：

Adam 优化器

lr: $5e-5$ ，weight decay: 0.1%，every 1000 steps

batchsize: 4，32 核心，实际 batch: 128

VSE++ 和 SCAN：

GRU 不是预训练编码器，直接使用 PhotoChat 来进行训练结果不会好，所以我们首先训练 GRU 模型在 MSCOCO 数据集并 finetune 他们在 Photochat 上 20 个 epoch。

Results of intent prediction

Table 2: Experimental results of the baseline models for the photo-sharing intent prediction task. All numbers are in percentage.

Model	F1 \uparrow	Precision \uparrow	Recall \uparrow
ALBERT-base	52.2	44.8	62.7
BERT-base	53.2	56.1	50.6
T5-base	58.1	58.2	57.9
T5-3B	58.9	54.1	64.6

Table 3: Number of negative turns and positive turns in each split of the dataset for the photo-sharing intent prediction task.

Split	Number of negatives	Number of positives
Train	68,795	10,286
Dev	6,802	1,000
Test	6,748	1,000

数据集标签不均衡，所以精度比较低。

Results of image retrieval

Table 4: Experimental results of the baseline models on image retrieval task. *DE* stands for our proposing dual encoders. *DE_{img}* only uses the image pixel values and *DE_{label}* only uses image labels to extract image features. *DE** is the model pretrained on MSCOCO. All numbers are in percentage.

Model	Loss function	R@1 \uparrow	R@5 \uparrow	R@10 \uparrow	Sum(R@1, 5, 10) \uparrow
BM25	-	6.6	15.4	23.0	45.0
DE _{label} (Bert-base)	CE	6.7	22.1	31.2	60.0
DE _{img} (ResNet-50)	CE	6.7	21.9	32.3	60.9
DE _{img} (ResNet-152)	CE	6.8	24.0	34.3	65.1
DE(ResNet-152, Bert-base)	CE	8.1	23.7	34.6	66.4
DE*(ResNet-152, Bert-base)	SH	8.0	22.0	31.0	61.0
DE*(ResNet-152, Bert-tiny)	SH	7.1	23.3	33.0	63.4
DE*(ResNet-152, Bert-base)	CE	8.5	26.1	35.3	69.9
DE*(ResNet-152, Bert-tiny)	CE	9.0	26.4	35.7	71.1
VSE++	MH	10.2	25.4	34.2	69.8
SCAN	MH	10.4	27	37.1	74.5

DE 是本文的 baseline，DElabel，DEimage 是消融实验。

CE 是交叉熵损失，SH 是 hinge loss，MH 是使用 hard negative 的 hinge loss。

如果 DE 模型在 MSCOCO 数据集上预训练然后在 PhotoChat 上 finetune，会有*标志。

在图像编码器上使用 ResNet-50 和 ResNet-152 比较；文本编码器使用 Bert-base 和 Bert-tiny 进行比较。

❖ 问题 1: 为什么没有不在 MSCOCO 上预训练的使用 Bert-base 的模型?

❖ 问题 2: 为什么 Bert-tiny 的效果反而比 Bert-base 好?

消融实验

使用图片的效果比单独使用图片标签的效果要好。(比较 DElabel 和 DEimage), 可能是图片包括了更多的信息。

effect of encoders

我们观察到使用小的模型去编码图像 label 效果要更好, 原因可能是标签是一个紧凑的 token 列表, 使用较小的模型可以缓解过拟合的问题。

另一方面, 使用一个更大的图像编码器 ResNet-152 可以产生更好的结果。

effect of loss function

交叉熵比 hinge loss 好。