Data poisoning attacks against multimodal encoders

贡献

1. 首个针对多模态模型的毒化攻击研究，包括语言和视觉模态毒化；
2. 三种毒化攻击，针对基于对比学习的多模态模型
3. 提出对文本和图像编码器的毒化攻击影响方式不同
4. 两种防御——预训练时和训练后防御

威胁模型：

- 攻击目标：模型 M，将有毒数据 D_p 注入到干净数据集 D_c 中形成训练集 $D = D_c \cup D_p$，在此训练集上训练有毒模型 M_p，目标是给定一些文本投毒模型能返回包含目标图像的列表。
- 攻击能力：攻击者能对训练数据投毒，但是因为互联网上公开数据很多，所以投毒率应该很低。其余是黑盒设置，也就是不知道目标模型的参数、架构等。

攻击重新

应该是数据投毒吧但是怎么看他只有目标没有具体的投毒方式啊？？？

攻击方法：

- 目标模型训练：训练集D、图像/文本数据 T/X
  训练数据 $\{(t,x)|(t,x) \in D = T \times X\}$，batch N，
  imgae encoder ε_{img}、text encoder ε_{txt}，
  -> 模型要最大化图像和文本embedding正对间的余弦相似度，同时最小化负对之间的距离。
- 文本-图像对 (t,x)，文本、图像 embedding $\mathcal{E}_t(t)$、$\mathcal{E}_i(x)$
- 交叉熵损失

$$L = -\sum_{1 \le i \le N} \sigma(\mathcal{E}_i(x_i), \mathcal{E}_t(t_i)) \cdot 1 - \sum_{1 \le i,j \le N} \sigma(\mathcal{E}_i(x_i), \mathcal{E}_t(t_j)) \cdot (-1)$$

  $\sigma()$ 是余弦相似度
- 攻击的主要思考点是怎么将数据投毒到干净数据集上，三种攻击：
  1. 单一目标图像：投毒比例 $\phi = \frac{|Dp|}{D}$，D_p 中每个投毒样本 $\{(t,x^*)|t \in T_A^{train}\}$，A是文本的原始类别，$T_A^{train}$ 代表在干净数据集上类别为A的文本子集，x^* 是属于不同类别的目标图像。
  2. 单一目标标签：原类别A→目标类别B，$\{(t,x)|t \in T_A^{train}, x \in X_B^{train}\}$
  3. 多目标标签：同时实现多个“单一目标标签”，目标$G = \{(A_1,B_1),(A_2,B_2),\ldots,(A_m,B_m)\}$

> 这里三种攻击只说了目标，所以他就是对数据集做了一个label的替换，将t类打一个错误的label x^* 就作为投毒数据集了啊

实验：

- 实验设置：使用预训练的 CLIP，图像encoder Vision Transformer ViT-B/32，文本 encoder用Transformer，然后在微调时进行投毒攻击。
- 数据集：Flickr-PASCAL、 COCO
- 投毒设置：

1. 攻击1：Flickr-PASCAL 中的“sheep”标记→一个“aeroplane”图像，COCO中的“boat”→一个“dog”图像，目标图像是从目标类别中随机选的
2. 攻击2：Flickr-PASCAL 和COCO中的 “sheep → aeroplane”和“boat→dog”。Flickr-PASCAL上毒化率0.08%，COCO上毒化率为0.24%
3. 共计3：为每个数据集设置两个目标，Flickr-PASCAL “sheep2aeroplane”和“sofa2bird”，COCO “boat2dog”和“zebra2train”

- 指标：
  - Hit@K：在图像/文本检索任务的排名列表的前K个实体中包含目标图像/文本的文本/图像样本的比例；
  - MinRank：所有测试图像的排名列表中目标图像的最小排名，MinRank越小意味着越早可以看到目标图像；
  - Cosine distance：取值范围0-2，相似则接近0。
- baseline：随机从测试集中选相同数量（和投毒数量一致）的文本数据，对这些进行检索并看结果。
- 实验结果：TR文本检索，IR图像检索

Table 1. Utility of poisoning attacks (Hit@10)

Dataset	Task	Clean	Attack I	Attack II	Attack III
Flickr-PASCAL	TR	0.984	0.980	0.980	0.958
	IR	0.971	0.973	0.968	0.954
COCO	TR	0.911	0.934	0.935	0.939
	IR	0.836	0.860	0.866	0.859

Table 2. Performance of Attack I

Dataset	Method	Hit@1	Hit@5	Hit@10	MinRank
Flickr-PASCAL	Baseline	0.000	0.032	0.032	79.168
	Ours	0.320	0.928	0.968	2.184
COCO	Baseline	0.000	0.020	0.036	153.852
	Ours	0.016	0.472	0.784	12.688

Table 3. Performance of Attack II

Dataset	Method	Hit@1	Hit@5	Hit@10	MinRank
Flickr-PASCAL	Baseline	0.024	0.088	0.200	51.048
	Ours	0.280	0.864	0.936	2.192
COCO	Baseline	0.024	0.072	0.116	123.076
	Ours	0.012	0.212	0.516	15.280

Table 4. Performance of Attack III

Dataset	Method	Hit@1	Hit@5	Hit@10	MinRank
Flickr-PASCAL	Baseline-1	0.048	0.120	0.216	46.576
	Ours-1	0.352	0.864	0.976	2.224
	Baseline-2	0.048	0.152	0.208	33.888
	Ours-2	0.008	0.248	0.552	12.792
COCO	Baseline-1	0.020	0.060	0.120	125.404
	Ours-1	0.016	0.272	0.604	13.940
	Baseline-2	0.012	0.020	0.032	288.496
	Ours-2	0.012	0.180	0.516	12.788

Figure 1. Cosine distance of the embeddings of the test samples between clean and poisoned models.

- 哪种模式更容易受到投毒攻击：

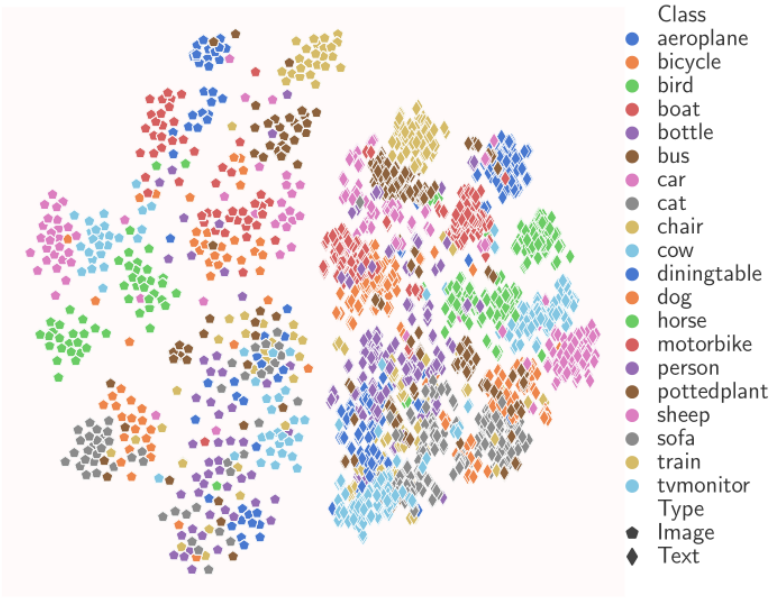

Figure 8. Embedding distribution of the PASCAL dataset.

图片embedding更稀疏，图像的embedding在毒化后变化会相对更大一点，也就意味着图像encoder更容易受到影响。

- 消融实验：相同毒化率下结果会随微调周期波动但整体有效；不同图像编码器对攻击效果影响不大；相同毒化率下攻击性能与数据大小无关；对person毒化目标会更难一点；对不同数据集也可迁移。

防御

两种防御：

- 训练前防御：数据防御，过滤可疑样本。relevance-文本和图像之间的余弦距离

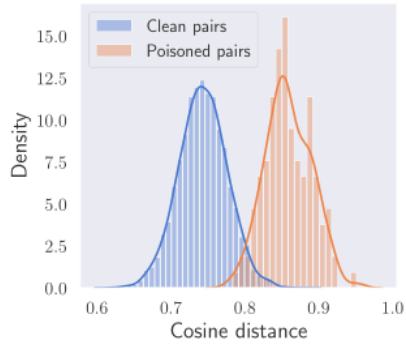

Figure 6. Probability density of cosine distances between clean/poisoned pairs in Flickr-PASCAL.

干净样本的余弦距离是0.75，有毒样本的余弦距离是0.85

> 这里依靠相似度的防御，直接通过构造相似度高的后门不就行了……

- 训练后防御：如果模型是被投毒过的，就用干净数据对其进行微调

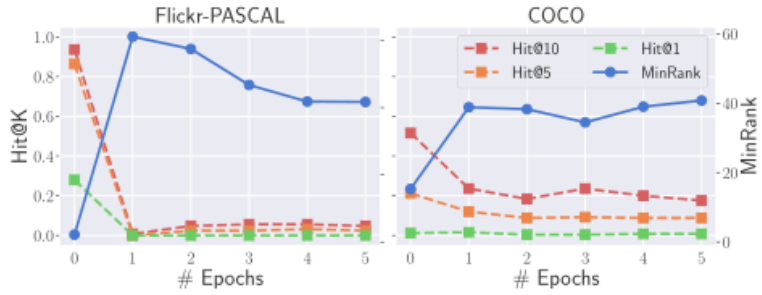

Figure 7. Performance of post-training defense against Attack II.

一轮epoch就防御明显

Table 7. Utility of post-training defense

Dataset	Hit@10 (TR)	Hit@10 (IR)
Flickr-PASCAL	0.978 (-0.006)	0.954 (-0.017)
COCO	0.976 (+0.065)	0.945 (+0.109)

Table 8. Influence of learning rate (LR)

Method	LR	Hit@1	Hit@5	Hit@10	MinRank
Attack II	-	0.280	0.864	0.936	2.192
Defense	10^{-3}	0.136	0.384	0.472	89.200
	10^{-4}	0.000	0.000	0.008	76.648
	10^{-5}	0.000	0.024	0.048	41.680

> 这里想到，干净数据微调很容易就把后门抹掉了，但是在此之前的攻击手段介绍那里，能凭很低的投毒率就能实现攻击，怎么感觉好夸张。

整篇文章感觉就比较base，更像是给出对多模态的攻击/防御框架，具体的攻击/防御手段并不新颖或者说根本就没详细解释，所以感觉就是最基本的方式去处理。