

目 录

1 研究目的	3
2 论文解读	3
2.1 注意力机制的介绍.....	3
2.1.1 注意力机制的原理.....	3
2.1.2 注意力机制运用的缺陷.....	4
2.2 AoA 方法与 AoANet 模型.....	5
2.2.1 AoA 方法	5
2.2.2 AoANet 模型——应用了 AoA 方法的图像描述模型.....	6
2.3 模型训练.....	9
2.3.1 数据集以及评价指标	9
2.3.2 训练方法	10
3 实验复现	11
3.1 实验流程.....	11
3.2 数据预处理.....	12
3.3 训练细节.....	12
3.4 实验结果与分析.....	12
3.4.1 定量分析.....	12
3.4.2 定性分析.....	13
3.4.3 消融实验.....	14
3.5 指定图像描述生成.....	16
3.6 困难及感悟.....	16
4 模型改进	17
5 结论	18
参考文献	19
附录 小组分工	20

1 研究目的

图像描述(Image Captioning)指的是计算机对于输入的图片,使用准确而有意义的句子,来描述图像所传达的重要信息。本质上来说,是计算机将提取的图像信息转换为更复杂的语义信息的过程。

研究者们提出了各种方法尝试完成此任务。Farhadi 等^[1]在 2010 年提出,在提取图像特征后,使用支持向量机对图像中可能存在的对象进行分类并生成描述;这种做法十分依赖于图像特征的提取和生成描述的规则,效果并不理想。2015 年,受机器学习用于文本翻译的编码器-解码器(Encoder-Decoder)框架^[2]的启发,谷歌团队^[3]将编码器常用的 RNN 更换为 CNN 后在图像描述任务上应用编码器-解码器框架,取得了较好的成果。在他们的基础上,Xu 等^[4]于 2016 年将注意力(Attention)机制融入了编码器-解码器框架中,进一步改进了效果。之后,许多研究者针对基于注意力机制的图像描述方法进行了改良。较为重要的改良包括 Lu 等^[5]提出的自适应注意力机制(Adaptive Attention)、Anderson 等^[6]提出的自底向上和自上向下相结合的注意力机制(Bottom-Up and Top-Down Attention),等等。

尽管对于图像描述任务,还可以使用其他方法如强化学习(Reinforcement Learning)^[7]、密集描述(Dense Captioning)^[8]等进行处理,但注意力机制的应用最为常见。此外,注意力机制还被广泛用于文本翻译等任务上,与它相关的研究和有价值、可供参考的成果数量可观。由于注意力机制的良好表现,我们希望进一步了解学习它的改进方法,以更好地完成图像描述任务。因此,我们小组选择了 ICCV2019 的文章 *Attention on Attention for Image Captioning*^[9]进行论文学习与复现。这篇文章提出了 AoA (Attention on Attention)方法,并通过在编码器-解码器框架上应用 AoA 方法,提高图像描述的效果。

2 论文解读

2.1 注意力机制的介绍

2.1.1 注意力机制的原理

深度学习中的注意力机制(图 1)与人的视觉注意力机制相似。人的视觉在快速扫描全局图像后获取了一些注意力焦点,并对焦点投入更多的精力,

从而在获取更多的有效信息的同时减少获取其它无效信息。这种视觉注意力机制可以大幅提高人处理信息的效率和准确性。深度学习中，注意力机制的目的也是提高信息提取的准确性和效率。

注意力机制将数据集划分为 **source** 和 **target** 两类（记为 **S** 和 **T**），把 **S** 中的数据整合为键值对<key,value>的形式，**T** 中的每个数据则是一个查询(query)。注意力机制的具体实现可以分为三个步骤：第一步是用 **T** 中的某个查询计算它与 **S** 中的每个键(key)的相似程度，可以使用的方法有求查询与键的向量点积、Cosine 相似性或引入其它神经网络进行求解；第二步是使用 **SoftMax** 等计算方法对结果进行归一化，同时更加突出重要元素的权重；第三步就是对这些数值进行加权求和得到最终结果，**SoftMax** 方法公式如下：

$$\alpha_{i,j} = f_{sim}(q_i, k_j), \alpha_{i,j} = \frac{e^{\alpha_{i,j}}}{\sum_j e^{\alpha_{i,j}}}$$

2.1.2 注意力机制运用的缺陷

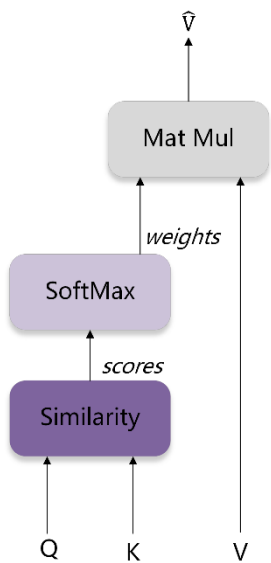


图 1：注意力机制示意图

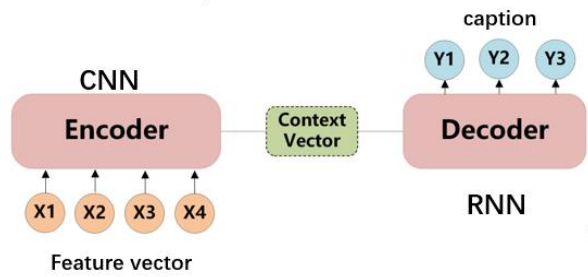


图 2：编码器-解码器框架

注意力机制虽然可以取得不错的结果，但在实际运用时也存在可以改进的地方。当注意力机制被运用于编码器-解码器框架（图 2）时，传统的做法是只在编码阶段加入注意力机制，针对不同的特征向量(feature vector)生成不同的语义编码(context vector)，这在一定程度上对重要信息进行了聚焦。但是解码阶段未施加注意力，处于类似“分心”的状态，因此无法判断先前的聚焦作用是否有效，如果生成的语义编码与原始信息的相关性很低，解码阶段依然会对这些语义编码进行解码，而结果可能与原始信息完全不相关。于是原文基于运用注意力机制的编码器-解码器框架设计了用于图像描述任务的 AoANet 模型，在传统注意力模块的输出上再施加一层注意力，对注意力机制加以改进。

2.2 AoA 方法与 AoANet 模型

2.2.1 AoA 方法

原文提出了 AoA (Attention on Attention)方法，通过明确经过注意力处理的结果(attention result)和查询之间的相关性，弥补注意力机制的缺陷。(图 3)

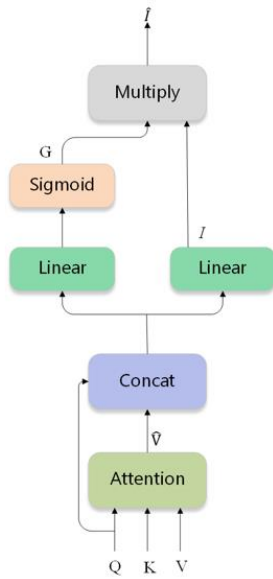


图 3: AoA 方法示意图

AoA 方法分为两个部分。第一部分对查询向量(q)和第一层注意力模块生成的结果向量(\hat{v} , 下文简称“结果向量”)进行线性变换，生成一个“信息向量”(i, information vector); 主要作用是保留第一次注意力的有效结果。AoA 方法的第二个部分生成一个“注意力门”(g, attention gate): 再次对查询向量

和结果向量进行线性变换，结果通过 **sigmoid** 函数进行激活，得到处理后的向量；这部分主要用于抛弃不必要的信息，比如查询和键值对之间没有相关性的部分。公式表达如下，其中， $W_q^i, W_v^i, W_q^g, W_v^g \in \mathbb{R}^{D \times D}$, $b^i, b^g \in \mathbb{R}^D$ ，分别为对信息向量和注意力门进行线性变换所需的参数。

$$\mathbf{i} = W_v^i \mathbf{q} + W_v^i \hat{\mathbf{v}} + b^i$$

$$\mathbf{g} = \sigma(W_q^g \mathbf{q} + W_v^g \hat{\mathbf{v}} + b^g)$$

然后，**AoA** 方法通过对信息向量和注意力门做逐位点乘，结合两个部分以添加第二层注意力，得到经过两次注意力的注意力信息($\hat{\mathbf{i}}$)。在模型中应用 **AoA** 方法时，注意力信息代替了结果向量被用于下一步操作中。

$$\hat{\mathbf{i}} = \mathbf{g} \odot \mathbf{i}$$

AoA 方法可以被表示为如下公式：

$$\begin{aligned} \text{AoA}(f_{att}, \mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \sigma(W_q^g \mathbf{Q} + W_v^g f_{att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + b^g) \\ &\odot (W_q^i \mathbf{Q} + W_v^i f_{att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + b^i) \end{aligned}$$

2.2.2 AoANet 模型——应用了 **AoA** 方法的图像描述模型

原论文沿用了大部分图像描述模型所使用的编码器-解码器框架搭建 **AoANet** 模型，在编码器和解码器模块都搭载了 **AoA** 模块进行改进。

首先，模型利用 **CNN (R-CNN)**为基础的网络结构对输入的图片进行处理，得到特征向量集 $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k\}$ ，其中 $\mathbf{a}_i \in \mathbb{R}^D$ 。**AoANet** 没有将这些特征向量直接放入解码器，而是构架了一个包含 **AoA** 模块的提炼网络(refining network)，由多个提炼模块(refining module)堆叠而成。提炼模块的结构如图 4 所示。提炼网络可以被看作 **AoANet** 的编码器模块。

提炼模块可以被表示为以下公式。

$$\mathbf{A}' = \text{LayerNorm}(\mathbf{A} + \text{AoA}^E(f_{mh-att}, W^{Q_e} \mathbf{A}, W^{K_e} \mathbf{A}, W^{V_e} \mathbf{A}))$$

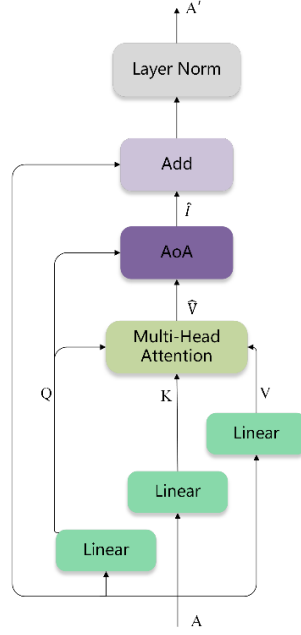


图 4: 精炼模块示意图

模型第一步将特征向量集 A 通过三个线性变换得到查询矩阵 Q 、键矩阵 K 、值矩阵 V ，并施加第一层注意力。文章基于多头注意力机制(Multi-head Attention)，即谷歌团队^[10]在 2017 年提出的自注意力机制(Self-attentive Multi-head Attention)，搭建了编码器的第一层注意力。

第一层注意力首先使用多头注意力机制方程(f_{mh-att})，将 Q, K, V 切割为 H 块(slice, 此处 $H=8$)，再对每一块分别应用放缩点积注意力机制方程($f_{dot-att}$, Scaled Dot-product Attention)，最终将结果整合为结果向量。

$$f_{mh-att}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_H)$$

$$\text{head}_i = f_{dot-att}(Q_i, K_i, V_i)$$

$$f_{dot-att}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d}}\right) V_i$$

随后，AoANet 将 AoA 方法应用于此结果向量和原有查询上，进行第二次注意力以得到注意力信息。对注意力信息与原特征向量集进行残差连接(residual connection)和层标准化(layer normalization)后，得到与原特征向量集维度大小一致的新特征向量集 A' 。由于维度大小一致，新特征集可以再次作为精炼模块的输入，实现精炼模块的堆叠以取得更好的效果。(本文中堆叠了 6 次)

精炼网络对特征向量间的关系进行了多次提炼，最终结果更好地表示了

图像中的关键信息。另外值得注意的是，AoANet 并没有完全按照多头自注意力机制设计第一层注意力：原模型中的前馈神经网络层(feed-forward layer)被 AoANet 所抛弃。这是由于，前馈神经网络层提供的非线性向量表达可以通过应用 AoA 方法实现；同时，去除前馈神经网络层并不影响模型的表现，反而使模型更加简洁。

在将特征向量集经过精炼网络处理之后，AoANet 基于自底向上和自上向下相结合的注意力机制搭建解码器（图 5）。在原模型中，解码器模块为“LSTM-注意力-LSTM”结构；AoANet 使用 AoA 方法代替了第二个 LSTM。

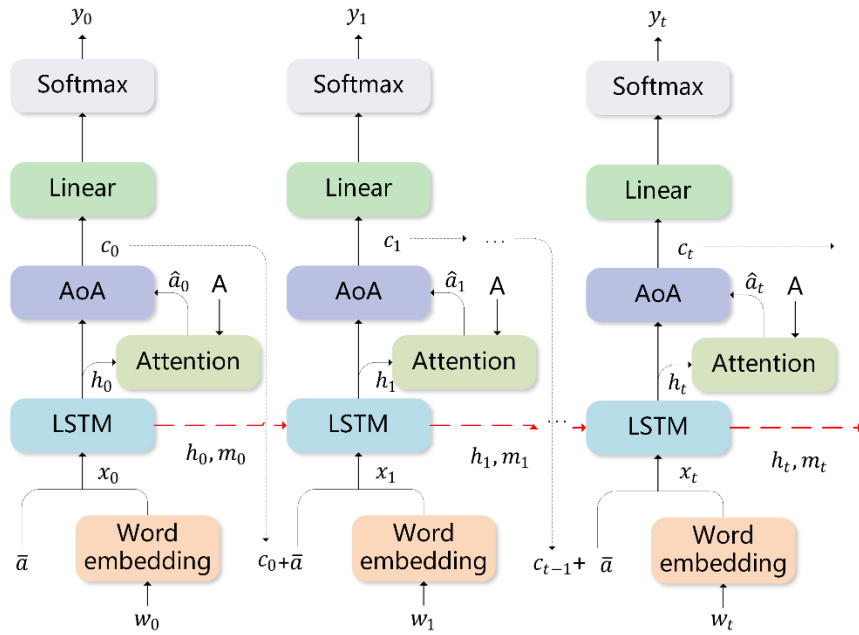


图 5：解码器模块示意图

在解码器模块，输入向量 x_t 由两个部分连接而成，一部分是将当前时间点的单词 w_t 经过独热编码得到的向量 Π_t ，进行词嵌入(word embedding)以后得到的向量；另一部分是可视向量(visual vector)，由特征向量集平均池化后得到的向量 \bar{a} 加上前一时间点的上下文向量 c_{t-1} 得到。当前时间点，LSTM 层解码后输出的隐藏层状态 h_t 和细胞状态 m_t 会被保留到下一时间点，与新时间点的输入向量再次经过 LSTM 解码。

$$x_t = [W_e \Pi_t, \bar{a} + c_{t-1}]$$

$$h_t, m_t = \text{LSTM}(x_t, h_{t-1}, m_{t-1})$$

输出的隐藏层状态会被传递到第一层注意力当中，作为查询向量，与编码器输出的特征向量集所变换构成的键值对进行第一次注意力，随后，隐藏

层状态和第一次注意力的结果向量放入 **AoA** 模块中第二次施加注意力，得到上下文向量 c_t ，保存了解码的状态和每个时间点新获得的信息。

$$c_t = \text{AoA}^D(f_{mh-att}, W^{Q_d}[h_t], W^{K_d}A, W^{V_d}A)$$

最后，通过计算上下文向量在词汇表上的条件概率，获得最终输出的结果 y_t 。

$$p(y_t | y_{1:t-1}, I) = \text{softmax}(W_p c_t)$$

2.3 模型训练

2.3.1 数据集以及评价指标

这篇文章使用的 MS COCO^[11]数据集拥有 123,187 张图像，该数据集起源于微软在 2014 年出资标注的 Microsoft COCO 数据集，主要用于解决目标检测，目标之间的上下文关系，目标的二维上的精确定位这三个问题。

该数据集的特点：

- 对象分割
- 在上下文中可识别
- 超像素分割
- 330K 图像
- 150 万个对象实例
- 80 个对象类别
- 91 个类别
- 每张图片有 5 个描述

文章对 MS COCO 数据集使用 “Karpathy”^[12]数据分割，分割出 5000 张图像作为验证集，5000 张图像作为测试集，剩余的图像作为训练集。

文章使用的评价指标如下，它们的结果越高表明模型的效果越好：

- **BLEU(Bilingual Evaluation Understudy)**: 这是一种基于准确率的评价指标，通过比较候选译文和参考译文里的 **n-gram**（文本中连续出现的 **n** 个词语）的重合程度来判断候选译文的质量，重合程度越高就认为译文质量越高。根据 **n-gram** 可以划分成多种评价指标，常见的指标有 **BLEU-1**、**BLEU-2**、**BLEU-3**、**BLEU-4** 四种，**BLEU-1** 衡量的是单词级别的准确性，而更高阶的 **BLEU** 可以衡量句子的流畅性。

- **METEOR(Metric for Evaluation of Translation with Explicit ORdering):** 这是一种基于 1-gram 的调和平均数和召回率的评价指标，也基于 F 值，通过计算特定的序列匹配，得到同义词、词根和词缀等之间的匹配关系，可以改善部分 BLEU 中的缺陷。METEOR 的结果与人工判断的结果有较高相关性。
- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation):** 是用于评估文本摘要算法的评价指标集合，基于召回率和 F 值。在图像描述任务中常用 ROUGE-L 指标，其中 L 代表最长公共子序列，ROUGE-L 计算了最长公共子序列的重合率。
- **CIDEr-D^[13] (Consensus-based Image Description Evaluation):** CIDEr 把每个句子看成文档，计算其 TF-IDF 向量的余弦夹角，据此得到候选句子和参考句子的相似度；CIDEr-D 在此基础上加以改进，可以更好地捕捉人类对“共识”的判断。
- **SPICE(Semantic Propositional Image Caption Evaluation):** 基于句子对应的语义场景图计算 F 值。

2.3.2 训练方法

文章首先使用交叉熵损失函数训练 AoANet，再使用 SCST^[14] (Self-critical Sequence Training)对模型进行优化。

交叉熵损失函数的原理是，在信息论中，熵用来表示一个事件所包含的信息量，事件发生的概率越小，蕴含的信息量就越大，并且独立事件的信息量可以叠加。所以熵的公式如下：

$$H(X) = -\sum_{i=1}^n p(x_i) \log(p(x_i))$$

其中 x_i 指不同的事件， $p(x_i)$ 指该事件发生的概率， n 代表 n 种可能。

而衡量两个分布之间的不同就需要使用 KL 散度，也称为相对熵，它表示的是对于同一个事件，用不同的两个分布刻画该事件所得到的信息增量，其计算公式如下：

$$\begin{aligned}
D_{KL}(p||q) &= \sum_{i=1}^n p(x_i) \log \left(\frac{p(x_i)}{q(x_i)} \right) \\
&= \sum_{i=1}^n p(x_i) \log(p(x_i)) - \sum_{i=1}^n p(x_i) \log(q(x_i)) \\
&= -H(X) + [-\sum_{i=1}^n p(x_i) \log(q(x_i))]
\end{aligned}$$

其中 $p(x_i)$ 表示分布 P 中事件 x_i 发生的概率， $q(x_i)$ 表示分布 Q 中事件 x_i 发生的概率， D_{KL} 的值越小，则两个分布越接近。可以看出等式的前一部分是事件的熵的相反数，等式的后一部分就是交叉熵。在使用 KL 散度评估真实值和预测值的差距时，熵的值不变，所以在优化过程中关注交叉熵即可。

而 SCST 优化了 CIDEr 评价标准。由于图像描述中生成单词的操作不可微，SCST 借鉴强化学习算法来训练网络，将贪婪搜索结果作为强化学习算法中的基线，而不需要用另一个网络来估计基线的值。这样的基线设置会迫使采样结果能接近贪婪搜索结果。在测试阶段，可直接用贪婪搜索产生图像描述，而不需要更费时的定向搜索(Beam Search)。SCST 的公式如下：

$$\begin{aligned}
L_{RL}(\theta) &= -\mathbf{E}_{\mathbf{y}_{1:T} \sim p_{\theta}}[r(\mathbf{y}_{1:T})] \\
\nabla_{\theta} L_{RL}(\theta) &\approx -(r(\mathbf{y}_{1:T}^s) - r(\hat{\mathbf{y}}_{1:T})) \nabla_{\theta} \log p_{\theta}(\mathbf{y}_{1:T}^s)
\end{aligned}$$

其中 $\mathbf{r}(\cdot)$ 是诸如 CIDEr-D 的评价指标， \mathbf{y}^s 是从概率分布中抽样的结果， $\hat{\mathbf{y}}$ 是贪婪解码的结果。

3 实验复现

3.1 实验流程

第一，掌握模型设计的原理和实现方式。我们阅读了原始论文和相关参考文献，学习了注意力机制、Transformer 架构等基础理论，理解了 AoA 机制的原理和 AoANet 模型结构。我们进一步研究了 AoANet 的开源代码，理清了完整的程序架构和运行机制，重点研读了模型的核心实现部分，对较为核心的文件进行了逐行梳理，确认其调用关系、代码上下文等，并与原文的理论部分进行了对应。

第二，实验设计。参考论文实验设置和分析、学习的需要，制定实验目标并规划实验内容。

第三，配置环境。在 Ubuntu 18.04 系统中配置 anaconda 环境，在 python3.6 虚拟环境中安装所需代码库；此前我们没有接触过 shell 脚本语言，因此还学习了如何通过命令行操作终端和执行程序。

第四，训练与评估模型。由于不熟悉 Pytorch 框架、深度学习实验经验欠缺等原因，我们在模型训练时遇到了较多问题，并逐一研究解决。完成 AoANet 模型训练后，又根据消融实验需要训练了其它模型。

最后，评估与应用。计算各模型的相关评估指标并分析。此外，还增加了可以针对指定图片生成字幕的功能，实现了对模型的加载和应用。

3.2 数据预处理

对于图像数据，使用基于 Resnet-101 的 Faster-RCNN 预训练模型处理，获得自底向上特征（bottom-up feature）^[6]。由于实验环境限制，该步骤直接采用开源预处理结果。原始的自底向上特征为 2048 维，通过映射将其长度转变为 1024。对于描述数据，将字母统一转化为小写并删除频数小于 5 的单词，得到含有 10369 个单词的词典。

3.3 训练细节

实验以交叉熵为损失函数迭代 25 次，使用 SCST 方法迭代 15 次，批量大小为 10，AoANet 训练总用时约 40 小时。针对交叉熵损失的训练初始学习率为 $2e-4$ ，SCST 优化初始学习率为 $2e-5$ ，并随迭代次数调整。解码器中，LSTM 隐藏层的结点数 1024。

3.4 实验结果与分析

参考原论文，进行多种实验评估 AoANet 的表现。

3.4.1 定量分析

		Bleu1	Bleu4	METEOR	ROUGE_L	CIDEr	SPICE
复现	Train	77.1	36.9	28.3	57.2	117.3	21.4
结果	Optimized	80.0	38.7	28.6	58.4	127.5	22.2
原文	Paper-train	77.4	37.2	28.4	57.5	119.8	21.3
结果	Optimized	80.2	38.9	29.2	58.8	129.8	22.4

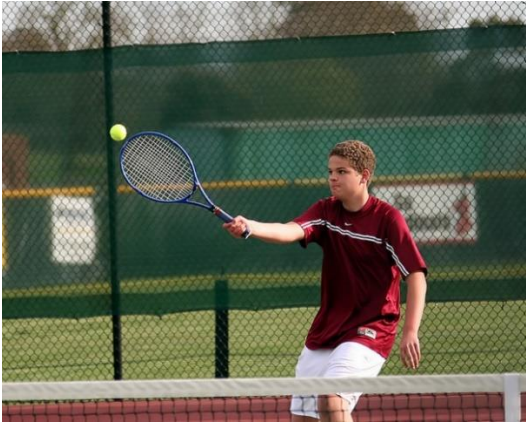
表 1：复现模型和论文模型得分比较

表 1 中,Train 为 AoANet 经过交叉熵优化的得分,Optimized 为经过 SCST 优化后的得分。本次复现与原文的结果相似,但略有出入,经分析有两个原因。第一,模型参数采用随机初始化,每次训练结果存在差异。第二,在作者开源代码的介绍中,针对交叉熵损失的迭代次数为 25 次,而在原文“实现细节”中的说明则是 30 次。实验中使用的迭代次数设置可能和作者汇报指标的实际设置不同。

参考原文中其它基线, AoANet 表现优异, 对原文的结论有所印证。

3.4.2 定性分析

针对模型输出,原作者认为 AoANet 可以生成不仅符合逻辑而且高质量的描述,相比其他模型具有的优势如下:一方面, AoANet 对同类元素的计数较为准确,如表 2 的第三个样本中有两只猫,第四个样本中有两块滑雪板;另一方面, AoANet 对图片中对象之间的语义关系更加敏感,在第一个样本中准确描述出了男孩正在用网球拍击球。我们训练了没有 AoA 模块的基线模型 (Base) 用于对比。分析 AoANet、Base 和没有经过 SCST 优化的 AoANet (AoA wo rl) 运行结果,还发现 AoANet 对于图中重要对象的识别能力更强,并验证了强化学习对模型的提升作用。这在样本二中有所体现。我们认为, AoANet 具有以上优势是因为 AoA 算法在编码-解码时利用信息向量关注有用的信息,利用注意力门抛弃不需要的信息,增强了注意力机制的效果。

图片	描述
	<p>AoANet: A young boy hitting a tennis ball with a tennis racket.</p> <p>AoA wo rl: A young boy hitting a tennis ball with a racquet.</p> <p>Base: A young boy holding a tennis ball on a tennis court.</p>



AoANet: A boat in the ocean with a mountain in the background.

AoA wo rl: A boat floating on top of a body of water.

Base: A boat is sitting in the ocean in a body of water.



AoANet: Two cats laying on a bed with a table.

AoA wo rl: A couple of cats laying on top of a bed.

Base: A black and white cat laying on a bed.



AoANet: A man sitting in the snow with two snowboards.

AoA wo rl: A person sitting on the snow with a snowboard.

Base: A person sitting in the snow with a snowboard.

注：AoANet、Base 经过 SCST 优化，AoA wo rl 未经过 SCST 优化

表 2：复现模型和论文模型得分比较

	Bleu1	Bleu4	METEOR	ROUGE_L	CIDEr
Base	76.2	34.9	27.1	56.1	110.3
+ Enc	77.5	36.7	28.1	57.1	117.0
+ Dec	76.7	35.8	27.9	56.8	114.8
AoANet	77.1	36.9	28.3	57.2	117.3

注：所有模型仅针对交叉熵损失训练

表 3：消融实验结果

3.4.3 消融实验

我们训练了一系列消融模型，用于检测 AoA 方法的效果。(见表 3)首先，我们以编码器、解码器均没有 AoA 模块的 Base 模型作为基线；其次，训练编码器中含有 AoA 模块，解码器中没有 AoA 模块的模型 (+ Enc)；编码器中没有 AoA 模块，解码器中含有 AoA 模块的模型 (+ Dec)。相比 Base 模型，在编码器和解码器中加入 AoA 模块都能提高模型的得分。经过对比，我们还发现编码器中的 AoA 模块对模型效果的提升作用更加显著。

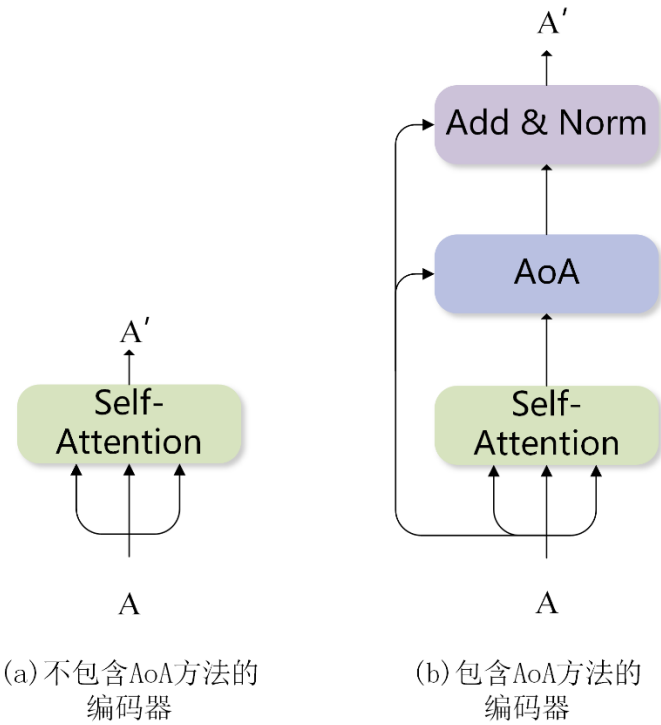


图 6: 编码器

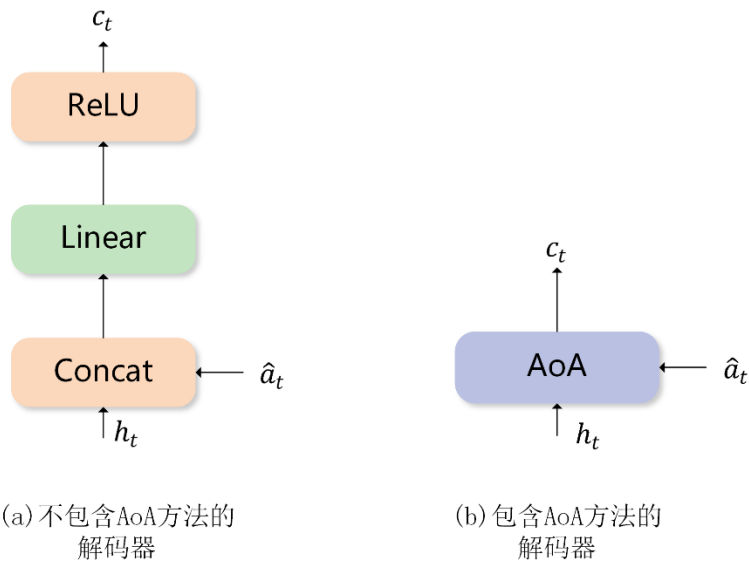


图 7: 解码器

3.5 指定图像描述生成

实验实现了加载模型，对指定图像生成描述。为方便使用，允许用户直接将图片放入指定目录，由程序分析目录内文件并建立图像列表。我们首先尝试了通过预训练模型本地抽取的方式获得图片特征，但使用的服务器账号没有管理员权限，配置预训练模型遇到困难；尝试各种方法后，将新图片的范围限定在 COCO 数据集中以获得图片特征。对于用户指定的每一张图片，程序找到对应特征，加载模型并输入特征，由模型生成描述。以表 2 中的图片为例，运行结果见图 7。

```
(pytorchenv) zs@inspur-1:~/aoaproject/AoANet-master$ bash generateCaption.sh
Captioning with model: log/log_base_rl/model.pth
image 0: a young boy holding a tennis ball on a tennis court
image 1: a boat is sitting in the ocean in a body of water
image 2: a black and white cat laying on a bed
image 3: a person sitting in the snow with a snowboard
Terminating BlobFetcher
Captioning with model: log/log_aonet_rl/model.pth
image 0: a young boy hitting a tennis ball with a tennis racket
image 1: a boat in the ocean with a mountain in the background
image 2: two cats laying on a bed with a table
image 3: a man sitting in the snow with two snowboards
Terminating BlobFetcher
```

图 8：图像描述生成结果

3.6 困难及感悟

小组成员在实际实验的过程中遇到了许多困难。有些困难是意料之中的，比如阅读大量级且结构复杂的代码、通过并不熟悉的 Linux 命令行操作远程服务器等；但更让人难以下手的是那些意料之外的问题，例如：原作者在 README 中提供的 python 和 pytorch 版本要求实际上并不适用于整个项目。记忆最深刻的是，在调试过程的中段，我们遇见了一个毫无头绪的报错，看遍了所有相关的代码部分都没找到任何语法或逻辑错误。最后我们根据报错信息，抱着一丝微缈的希望查看了出错的函数在新旧版本 pytorch 中的源代码，然后惊奇地发现：在新版本 pytorch 中，该函数有两个参数交换了位置，而这个交换正好可以解释出现的错误。于是我们怀着激动又紧张的心情，在调用函数时改变了传参方式，然后运行代码。看到这部分没有再报错，两个人爆发出几乎是喜极而泣的欢呼，停不下来的大笑长达一分钟，然后继续解决这次运行中出现的新问题。

在充满挑战的同时，本次项目也是一次很愉快的体验。解决上述困难的途中，我们收获了解决困难 bug/完成从未完成过的任务时的成就感、论文和代码的阅读能力、对高阶深度学习模型的理解等等。此外，小组成员均认真且有质量地完成了自己负责的部分，四个人也都愿意为之付出足够的时间——许多个没有晚课的日子里，我们在结束了白天的课程后一起坐在图书馆的研讨室，共同度过从 6 点到 10 点闭馆的夜晚。更进一步地，我们在充分理解并完成原始代码的基础上拓宽了实验范围，如利用训练得到的模型，设计新的程序框架来读入指定图像特征并生成描述；以及通过消融实验分析模型设计在实际应用中起到的效果。诚然，这些附加实验要求我们投入一定的时间，还让我们遇到了更多始料未及的状况；我们一度怀疑是否应该减少一些实验项目，也确实因为环境等原因有所妥协。但最终我们基本实现了完善实验的设想，在这个过程中学习到了各方面的知识，并对模型有了更深刻的理解。

4 模型改进

AoA 方法利用信息向量和注意力门来加强注意力机制的效果，得到较好结果。不过，我们在实验和查阅资料的过程中发现其仍然存在一些缺陷，注意力机制也可以进一步优化。以下是部分较为前沿的改进方法。

Cornia 等^[15]提出了 \mathcal{M}^2 模型——网状内存注意力机制模型(Meshed-Memory Transformer)：通过在 Transformer^[16]模型上添加内存向量，利用内存向量和先验知识更好地关注到图片中的细节，从而达到比 AoANet 更好的注意力效果。Pan 等^[16]从不同的角度出发：他们分析传统的注意力机制，发现它们通常只挖掘了不同模态之间一阶的特征交互信息(feature interaction)；因此他们提出了 X 线性注意力机制(X-Linear Attention)，来发掘不同模态间更高阶的特征交互信息，以获得更好的结果。

而 Zhang 等^[17]则指出，由于 Transformer 模型本身是基于文本翻译被提出的，因此直接将自注意力机制应用于图像(如 AoANet 的提炼网络部分)，会导致图像的一些位置信息丢失，同时，图像并不能够传达一些语法必需的、与图像无关的词语，所以需要另一个语言标志来帮助生成描述；他们设计了网格增强模块(Grid-Augmented Module)来改善视觉信息的表示，并在 Transformer 模型基础上

提出了应用于解码器的适应注意力机制(Adaptive Attention)来关注位置信息，语言标志则通过 BERT 来获得。同样关注图像的位置信息，Zhong 等^[18]则利用子网格(sub-graph)来分割图片，进而得到复杂图片对象之间的关系。

另外，Wang 等^[19]指出，包括 AoANet 在内的、现有的应用于图像描述上的注意力机制模型在训练时存在偏误，它们更偏好于普遍性的描述，而忽略了图像中存在的特殊信息。他们提出了一个新的评价指标 SPICE-U 和用于改进现有注意力机制的方法，在解码器中同时关注 $P(\text{Sequence}|\text{Image})$ 和被现有模型忽略的 $P(\text{Image}|\text{Sequence})$ ；在新的评估方法参与训练时，AoANet 可以给出更好的注意力结果。

5 结论

本次对论文 *Attention on Attention for Image Captioning* 的复现任务基本完成。原文认为编码器-解码器结构中的解码器缺乏对注意力机制输出和查询关系的信息，提出了 AoA 方法弥补这一缺陷；据此设计的模型 AoANet 提升了图像描述的效果。我们复现了 AoANet 模型，验证和分析其在数据集上的表现，并在学习此论文和相关论文的过程中对图像描述任务、注意力机制等知识有了更深刻的认识。

参考文献

- [1] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier and D. Forsyth, "Every Picture Tells a Story: Generating Sentences from Images," in ECCV, 2010.
- [2] K. Cho, B. v. Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," Statistics, 2014.
- [3] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and Tell: A Neural Image Caption Generator," in CVPR, 2015.
- [4] K. Xu, J. L. Ba, K. C. Kiros, A. Courville, R. Salakhutdinov, R. S. Zemel and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," in ICML, 2015.
- [5] J. Lu, C. Xiong, D. Parikh and R. Socher, "Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning," in CVPR, 2017.
- [6] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould and L. Zhang, "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering," in CVPR, 2018.
- [7] S. Liu, Z. Zhu, N. Ye, S. Guadarrama and K. Murphy, "Improved Image Captioning via Policy Gradient optimization of SPIDeR," in ICCV, 2017.
- [8] L. Yang, K. Tang, J. Yang and L.-J. Li, "Dense Captioning with Joint Inference and Visual Context," in CVPR, 2017.
- [9] L. Huang, W. Wang, J. Chen and X.-Y. Wei, "Attention on Attention for Image Captioning," in ICCV, 2019.
- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention Is All You Need," in NeurIPS, 2017.
- [11] Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In ECCV, 2014.
- [12] Andrej Karpathy and Fei Fei Li. Deep visual-semantic alignments for generating image descriptions. In CVPR, 2015.

- [13] R. Vedantam, C. L. Zitnick and D. Parikh, "CIDEr: Consensus-based Image Description Evaluation," in CVPR, 2015.
- [14] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross and V. Goel, "Self-critical Sequence Training for Image Captioning," in CVPR, 2017.
- [15] M. Cornia, M. Stefanini, L. Baraldi and R. Cucchiara, "Meshed-Memory Transformer for Image Captioning," in CVPR, 2020.
- [16] Y. Pan, T. Yao, L. Yehao and T. Mei, "X-Linear Attention Networks for Image Captioning," in CVPR, 2020.
- [17] X. Zhang, X. Sun, Y. Luo, J. Ji, Y. Zhou, Y. Wu, F. Huang and R. Ji, "RSTNet: Captioning with Adaptive Attention on Visual and Non-Visual Words," in CVPR, 2021.
- [18] Y. Zhong, L. Wang, C. Jianshu, D. Yu and Y. Li, "Comprehensive Image Captioning via Scene Graph Decomposition," in ECCV, 2020.
- [19] Z. Wang, F. Berthy, K. Narasimhan and O. Russakovsky, "Towards Unique and Informative Captioning of Images," in ECCV, 2021.

附录 小组分工

小组成员共同完成了选题、文献调研、原论文和核心代码的学习、项目报告的完善，其余事项分工如下：

胡语晴&章姝：阅读研究原始代码，梳理代码结构框架，将论文模型设计与核心代码实现进行对应学习。配置远程 Linux 服务器环境并训练模型完成多项实验，整理分析实验结果，在原始代码的基础上还进行了进一步拓展应用。另负责了代码 README 的撰写以及报告和项目展示的实验部分；

陈逸云&顾萌桢：学习研究原论文，并进一步学习原论文中涉及的相关基础知识，梳理知识发展背景和文章结构。基于参考文献对原论文中的模型进行分析，整理模型改进方法。负责报告和项目展示的理论部分。