Recent VQA Approaches

Wentao Mo[1]

[1]Department of Machine Intelligence
Peking University

2021 年 4 月 29 日

Outline

1. Visual Feature Extractor
   - VC R-CNN (CVPR '20)
   - Grid Feature (ECCV '20)
2. Feature Fusion Methods
   - MFH (TNNLS '18)
   - MCAN (CVPR '19)
   - TRRNet (ECCV '20)
   - BERT/Transformer-based Fusion
     - UNITER (CVPR '20)
     - OSCAR (CVPR '20)

Visual Commonsense R-CNN

提出 VC R-CNN. 使用 causal intervention$P(Y \mid \mathrm{do}(X))$ 代替传统的 IId.
相信应该使用 causual commonsense feature 而不是单纯的 visual feature.
Replace

$$P(Y \mid X) = \sum_z P(Y \mid X, z)\underline{P(z \mid X)} \tag{1}$$

w/ intervention

$$P(Y \mid do(X)) = \sum_z P(Y \mid X, z)\underline{P(z)} \tag{2}$$

提出 proxy task 为预测 local context label of Y. 关于 confounder set Z, 我们保存一些固定数量的 dictionary $N \times d$, N 是数据集中类别的数量 (MSCOCO, 80), 每个 d 维特征都是平均的 RoI 特征. 特征通过 Faster RCNN pretrain.
总 obj. 为 self-classification+contextual-pair-classification loss

$$L(X) = L_{\text{self}}\ (p, x^c) + \frac{1}{K}\sum_i L_{cxl}\left(p_i, y_i^c\right) \tag{3}$$

Visual Commonsense R-CNN

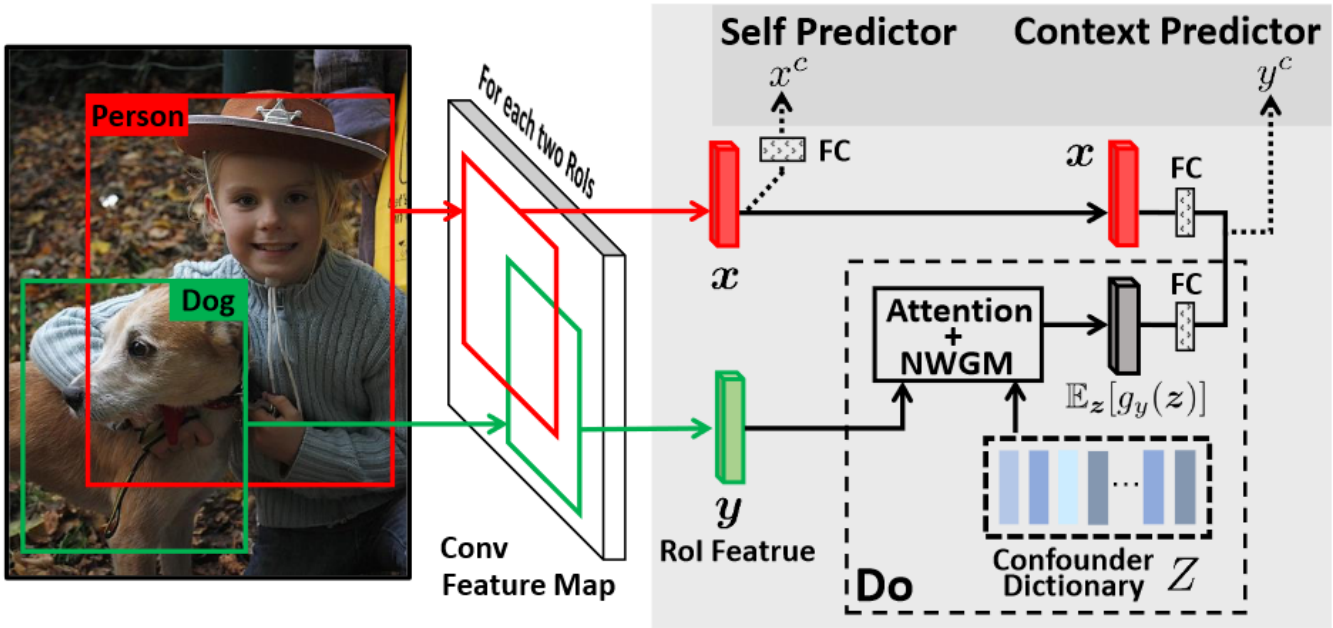


Figure 4. The overview of VC R-CNN. Any R-CNN backbone (*e.g.*, Faster R-CNN [54]) can be used to extract regions of interest (RoI) on the feature map. Each RoI is then fed into two sibling branches: a **Self Predictor** to predict its own class, *e.g.*, x^c, and a **Context Predictor** to predict its context labels, *e.g.*, y^c, with our **Do** calculus. The architecture is trained with a multi-task loss.

Visual Commonsense R-CNN

具体上, $P(Y \mid do(X)) = \sum_z P\left(y^c \mid \boldsymbol{x}, \boldsymbol{z}\right) P(\boldsymbol{z})$, 使用

$$P(Y \mid do(X)) := \mathbb{E}_{\boldsymbol{z}}\left[\text{Softmax}\left(f_y(\boldsymbol{x}, \boldsymbol{z})\right)\right] \overset{\text{NWGM}}{\approx} \text{Softmax}\left(\mathbb{E}_{\boldsymbol{z}}\left[f_y(\boldsymbol{x}, \boldsymbol{z})\right]\right) \tag{4}$$

并且使用 NWGM(Normalized Weighted Geometric Mean) 来估计上述期望
f 使用线性模型 $f_y(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{W}_2 \cdot g_y(\boldsymbol{z})$, where $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{N \times d}$ 代表了 FC 层. 那么有

$$\mathbb{E}_{\boldsymbol{z}}\left[f_y(\boldsymbol{x}, \boldsymbol{z})\right] = \boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{W}_2 \cdot \mathbb{E}_{\boldsymbol{z}}\left[g_y(\boldsymbol{z})\right] \tag{5}$$

建模 $g_y(\cdot)$ 为 scaled 点积注意力, 具体上有

$$\mathbb{E}_{\boldsymbol{z}}\left[g_y(\boldsymbol{z})\right] = \sum_z \left[\text{Softmax}\left(\boldsymbol{q}^T \boldsymbol{K} / \sqrt{\sigma}\right) \odot \boldsymbol{Z}\right] P(\boldsymbol{z}) \tag{6}$$

使用 NCC 去除 $x \to z$ 的样本.

In Defense of Grid Features for VQA

最近基于 region 的方法逐渐流行并超过了基于 grid 的方法. 但是相容实验发现主要影响性能的是 pre-training 的数据集质量 + 输入图像的高分辨率, grid/region 只是小问题.
传统上, 一般使用 Faster R-CNN, 在 cleaned version VG 上训练. 对于这些方法, 要获得自底向上注意力特征, 进行如下两步

1. Region Selection. 通过一个 Region Proposal Net., 提出候选 region(Regions of Interst, RoIs), 接着通过一个 score comp., 选择 top-N 的区域, 并且两步都使用 NMS.
2. 给出了上述步骤的 regions, 使用 RoIPool 来得到 region-feature.

由于 VG 数据集的复杂性和 Faster R-CNN, 这两步计算上都很昂贵.
具体的 VQA 模型 (特征融合) 使用的是 MFH.

In Defense of Grid Features for VQA

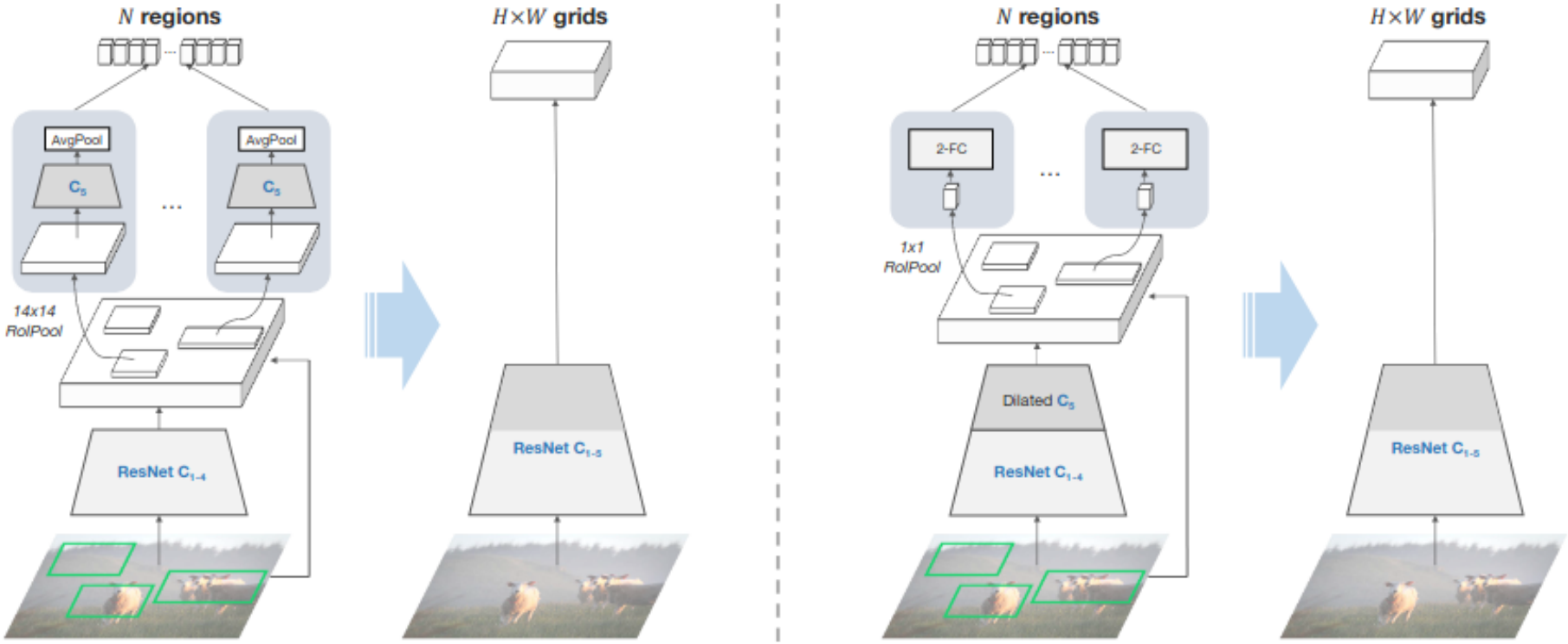


Figure 2: From regions to grids. Left: We convert the original region feature extractor used by bottom-up attention [2] back to the ResNet [15] grid feature extractor for the *same* layer (see Sec. 3.2, weights in blue are transferred), and find it works surprisingly well for VQA [11]. **Right**: We build a detector based on 1×1 `RoIPool` while keeping the output architecture *fixed* for grid features (see Sec. 3.3), and the resulting grid features consistently perform at-par with region features.

In Defense of Grid Features for VQA

Faster R-CNN 是 c4 模型 w/ 属性分类分支的变种. 首先使用 ResNet 的 C_4blocks 来得到 feature map, 接着 per-region feature 先 14×14 RoIPool, 再应用 C_5, 最后 avg-pool 来得到每个 region 的 F. 我们直接使用 C_5 在 grids 来得到特征.
这意味着用一个一维向量来表示一个 region, 而不是 Faster RCNN 里的 HWC 三维. 使用 1×1 RoIPool 会降低物体检测的性能, 对于 VQA, 这要求这个特征尽可能单独地编码信息. 由于预训练的 C_5 输入不适用, 使用最近的直接使用整个 C_5 的 ResNet 的工作[1].
Ablation Study 发现主要影响性能的是 pre-training 的数据集质量 + 输入图像的高分辨率, grid/region 只是小问题. 使用 Res-NeXt 改进了性能. 发现使用更大的图像, 更高的精度. 不同的预训练任务上, detection w/ attr. $>$ detection w/o attr. $>$ classification w/ tag $>$ cls. w/ label.

[1] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. De-formable convnets v2: More deformable, better results. InCVPR, 2019.

MFH: Multimodal Factorized High-order Attention

直觉上, 使用注意力机制进行特征融合是很自然的. 所以他们使用 co-attention 模块来同时学习两边的 attention, 并且使用低秩估计计算双线性特征融合.
对于视觉特征 $x \in \mathbb{R}^m$ 和语言特征 $y \in \mathbb{R}^n$ 有低秩估计二次型

$$\begin{aligned} z_i &= x^T U_i V_i^T y = \sum_{d=1}^{k} x^T u_d v_d^T y \\ &= 1^T \left(U_i^T x \circ V_i^T y \right) \end{aligned} \tag{7}$$

, 实现上可以写成

$$z = \text{SumPool}\left(\tilde{U}^T x \circ \tilde{V}^T y, k \right) \tag{8}$$

之后接上 Dropout + Power Normalization($z \leftarrow \text{sign}(z)|z|^{0.5}$) + ℓ_2 Normalization ($z \leftarrow z/\|z\|$)

MFH

基于单层 MFB, 提出多层的 MFH 结构, 设 MFB 里的中间表示为

$$z_{\exp} = \mathrm{MFB}_{\exp}(x, y) = \mathrm{Dropout}\left(\tilde{U}^T x \circ \tilde{V}^T y\right) \in \mathbb{R}^{ko} \tag{9}$$

以及

$$z = \mathrm{MFB}_{sqz}\left(z_{\exp}\right) = \mathrm{Norm}\left(\mathrm{SumPool}\left(z_{\exp}\right)\right) \in \mathbb{R}^{o} \tag{10}$$

则有

$$z_{exp}^{i} = \mathrm{MFB}_{exp}^{i}(x, y) = z_{exp}^{i-1} \circ \left(\mathrm{Dropout}\left(\tilde{U}^{i^T} x \circ \tilde{V}^{i^T} y\right)\right) \tag{11}$$

其中第 0 层 $z_{exp}^{0} \in \mathbb{1}^{ko}$ 最后 concat 各层特征

$$z = \mathrm{MFH}^{p} = \left[z^1, z^2, \ldots, z^p\right] \in \mathbb{R}^{op} \tag{12}$$

MFH

视觉 backbone 是 ImageNet 上预训练的 ResNet-152, 语言特征上使用的是 LSTM. 进行 co-attention+MFH 融合

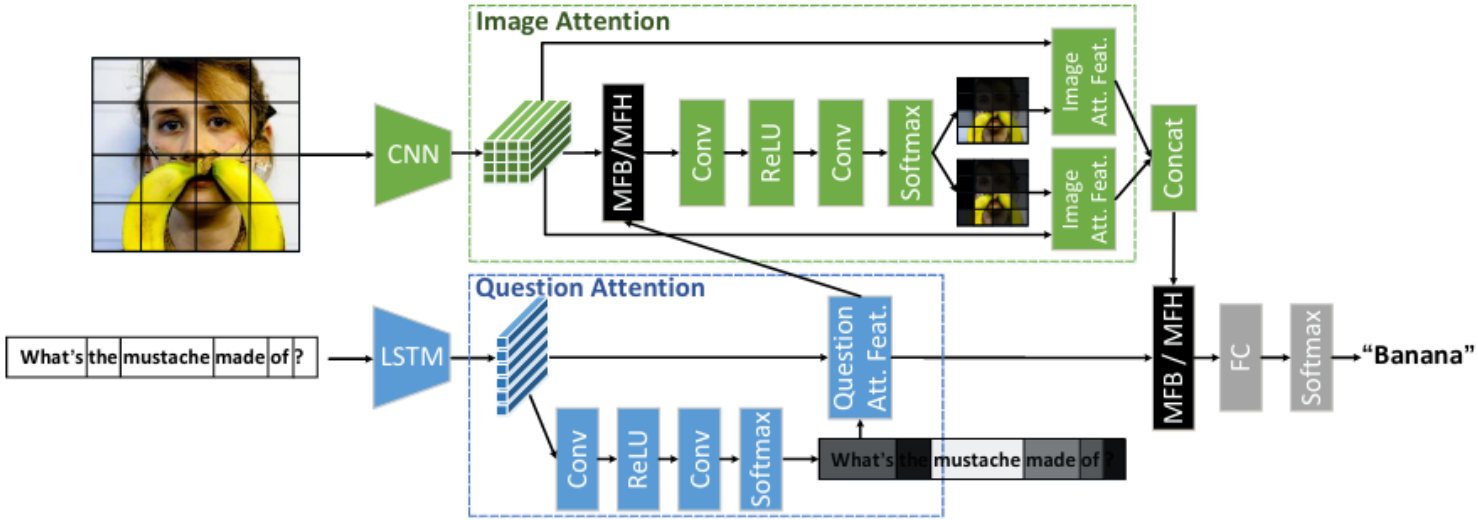


Fig. 5. The co-attention network architecture with MFB or MFH for VQA. Different from the network of MFB baseline, the images and questions are firstly represented as the fine-grained features respectively. Then, *Question Attention* and *Image Attention* modules are jointly modeled in the framework to provide more accurate answer predictions. For both the image and question attention modules, multiple attention maps (see the example in the image attention module) can be adapted to further improve the representation capacity of the fine-grained features.

相容实验中的发现:

1. 缺少 ℓ_2 norm. 降低了性能 (-3%), power norm. 影响并不大.
2. ℓ_2 norm. 明显让 neuron value 变得稳定, 让模型更稳定.

Remark 这里 ℓ_2 正则化是不是类似于 LayerNorm?

MCAN: Deep Modular Co-Attention Networks

本工作设计了一种好的 co-attention 机制来做 VQA 任务的特征融合. 使用 SA(SelfAtt) 模块建模 intra-modality att 和 GA(Guided Att) 模块来建模 inter-modality att. 通过结合它们, 得到了 MCA 层.

Remark Commonsense 推理 (如同 VC RCNN 中) 是否在 coattention/correlation inference 里 make sense?

视觉特征: 使用 10-100 个目标区域 (取决于 confidence, w/ threshold), $X \in \mathbb{R}^{m \times d_x}$.
语言特征: 问题 token 化, 转换成最多 14 个词, 每个 word 使用 300d GloVe(pretrained) embed. 词 embed. 送到单层 LSTM 中, 并且使用整个输出作为问题的特征矩阵 $Y \in \mathbb{R}^{n \times d_y}$.

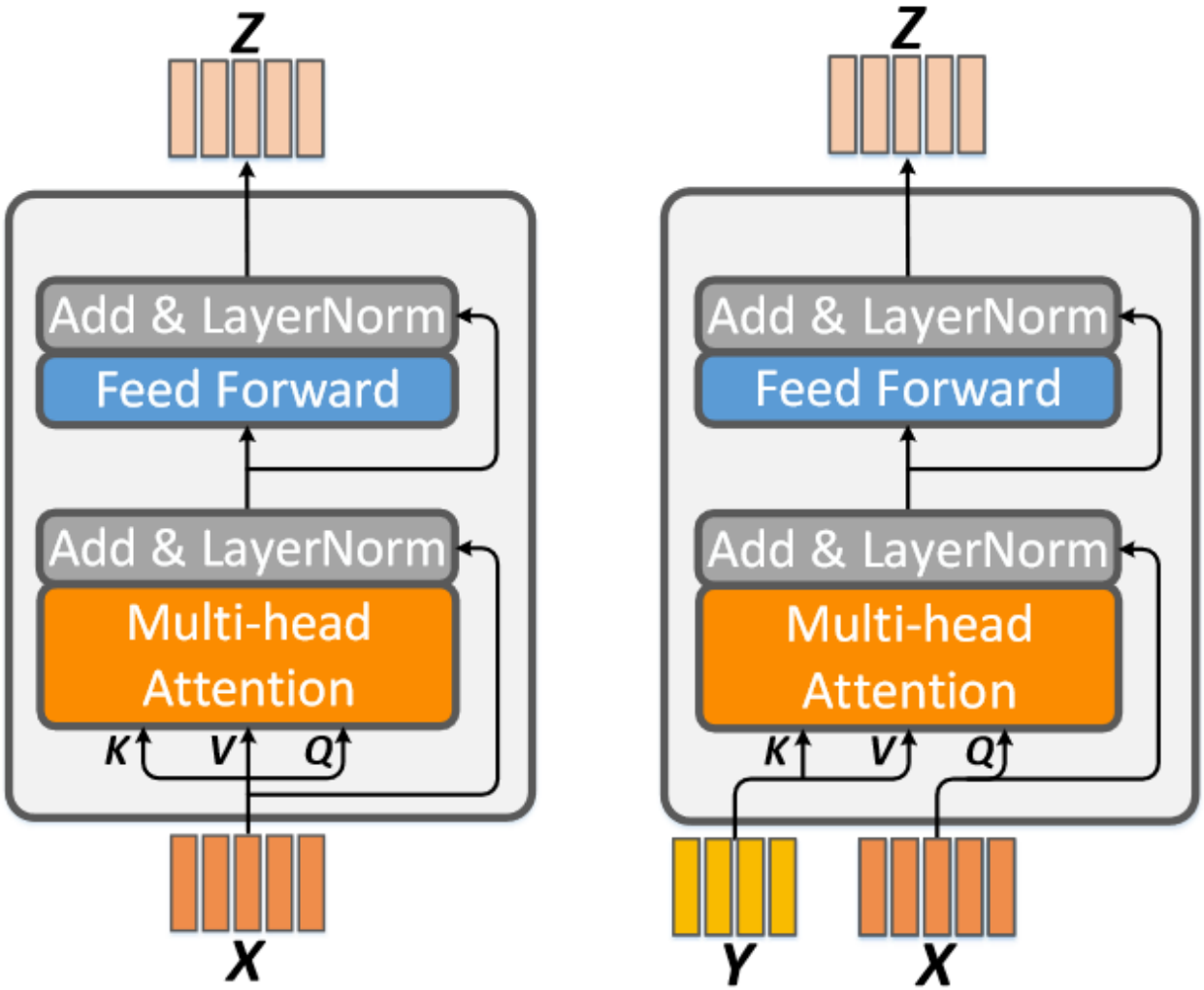
MCAN

为了处理可变长长度, 使用 0-padding 来把填充到最长.(i.e.,m=100, n=14). 每个 softmax 之前把 padding logits 改成 $-\infty$.
最后, 加上 attentional reduction(FC(d)-ReLU-Dropout(0.1)-FC(1)) 来得到视觉特征注意力, 然后融合

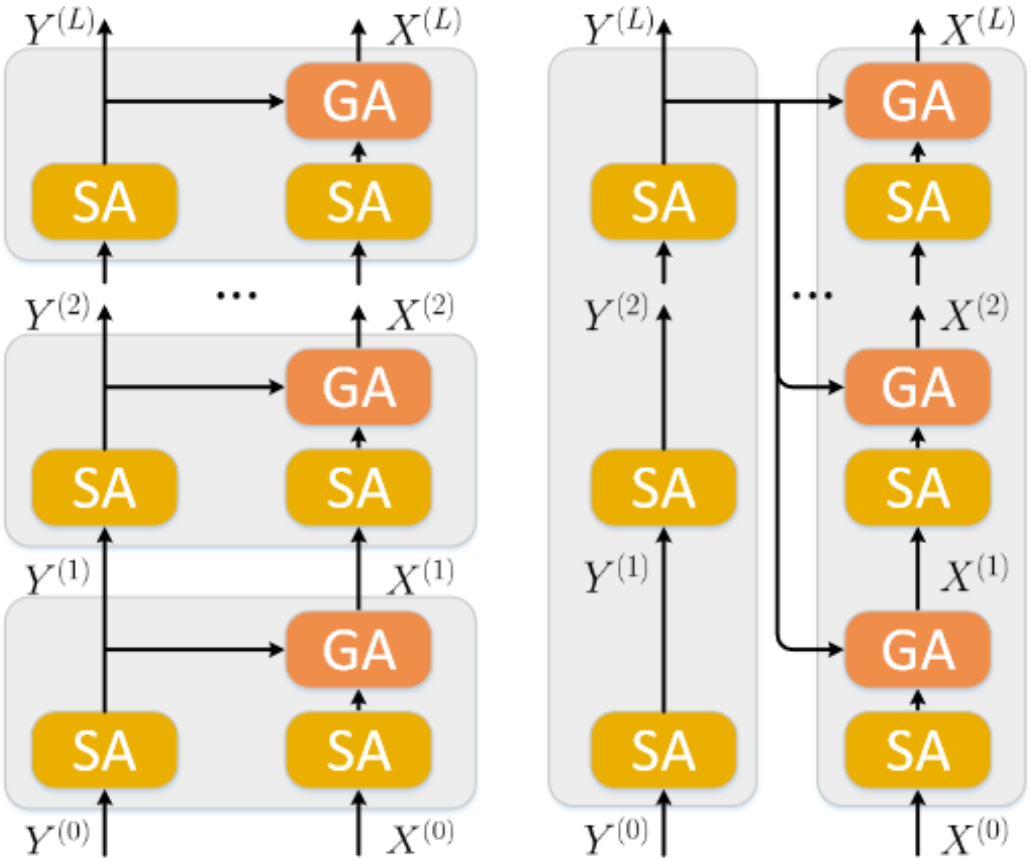
$$z = \text{LayerNorm}\left(W_x^T \tilde{x} + W_y^T \tilde{y}\right) \tag{13}$$

送到线性 N-classifier 上.
最后使用 $MCAN_{ed} - 6$ 作为 best single model, 精度 70.63/70.90 (test-dev/std), 在 train+val+vg 上训练.(vg 是从 VG 数据集来的增强 VQA 样本!). 通过 ensemble, 精度可以达到 72.45.

MCAN


Z
Add & LayerNorm
Feed Forward
Add & LayerNorm
Multi-head Attention
K
V
Q
X
Z
Add & LayerNorm
Feed Forward
Add & LayerNorm
Multi-head Attention
K
V
Q
Y
X

MCAN



(a) Stacking (b) Encoder-Decoder

Figure 5: Two deep co-attention models based on a cascade of MCA layers (*e.g.*, SA(Y)-SGA(X,Y)).

TRRNet: Tiered Relation Reasoning for Compositional VQA

本工作提出 tiered relational reasoning 来进行逐步推理/合成推理, 每步只在动态选择的对象 candidates 之间学习关系. 每个 TRR unit 包含:root attention, root to leaf attention, leaf attention, 为了传递给下一个 unit 的 message passing module.
视觉输入经过 OD, 提取出 region 视觉特征, $V \in \mathbb{R}^{n \times d_v}$, 和 bounding box 特征 $B \in \mathbb{R}^{n \times d_b}$. 语言经过 BERT 一样的词嵌入, 并且送进 GRU 来得到更好的语言 embedding $E \in \mathbb{R}_e^{m \times d}$

TRRNet: TRR Unit

1. **Root Attention** 对象级别注意力

$$\alpha^{\text{object}} = \text{softmax}(\text{Net}(V, B, E)) \tag{14}$$

以及 attended feature

$$O^{\text{root}} = \alpha^{\text{object}} V^T , \tag{15}$$

本作使用 Bottom-Up 和 BAN 里的注 bt 意力方法.

2. **Root to Leaf Attention Passing** 使用 multi-head **hard** att. 来选择 rel. obj. 候选. 具体上, 我们通过选择每个 att. head 的 top-k attended obj., 再通过 concat+MLP 映射到 $d_r (= 256 \text{ in this work})$, 表示为 Relation 操作:

$$\begin{aligned} V_{\text{Hard}} &= \text{Topk}(\alpha, V, K) \\ R &= \text{Relation}\ (V_{\text{Hard}}, B), \end{aligned} \tag{16}$$

TRRNet: TRR Unit

- **Leaf Attention** 和 root att. 相同, 使用关系表示 $R \in \mathbb{R}^{K^2 \times d_r}$ 和问题 embedding $e \in \mathbb{R}^{d_e}$

$$h = f(g(e) \odot k(R)), \tag{17}$$

其中 g,k,f 是 fc(ReLU).

$$\alpha^{\text{relation}} = \text{softmax}(h) \tag{18}$$

以及

$$O^{\text{leaf}} = \alpha^{\text{relation}} R^T . \tag{19}$$

- **Message Passing module for units interaction** 为了可以进行 multi-stage reasoning, 提出 message passing module,

$$V_{new} = f\left(\left[O^{leaf}, V\right]\right) \tag{20}$$

TRRNet: Multi-stage Reasoning and Policy Network

Cascade 多个 TRR Unit

$$O_t^{\text{root}}, O_t^{\text{leaf}}, V_{t+1} = TRR_t(B, V_t, E) \tag{21}$$

使用决策网络来决定 (部分取决于 att. feat. 输出是否有区别) 是否进行下一个 TRR Unit 的推理

$$\begin{aligned} d &= L2\left(O_{t-1}^{\text{root}}, O_t^{\text{root}}\right) \\ z &= MLP[MLP(d), MLP(E, t, l)], \\ \pi(a_t \mid s_t, \theta) &= \text{softmax}(z) \end{aligned} \tag{22}$$

使用 REINFORCE 中的 policy gradient 方法来训练. 在训练中分别地训练两个网络 (同时保持另一个网络参数固定, 并且最多三步), policy net. 的 loss 是

$$L = -E_{s\sim\pi}[r - p] \tag{23}$$

其中每走一步 penalty 为 0.1.

TRRNet: The Readout Layer

最后如下 Readout, concat 各层 feature,

$$O^{\text{all}} = f\left(\left[O_t^{\text{root}}, O_t^{\text{leaf}}\right]\right) \tag{24}$$

$$E^{\text{final}} = g(E) \tag{25}$$

$$\text{Answer} = \text{softmax}\left(h\left(O^{all} \odot E^{final}\right)\right), \tag{26}$$

最后相容实验指出, 带 PN 比硬编码要提高了 0.5%.

UNITER: Transformer-like Fusion and Pretraining

Pre-trained 语言模型大大改进了 NLP 的能力 (ELMo, BERT, GPT-2, XLNet, RoBERTa, ALBERT), 特点是在大型数据集上进行预训练, 同时使用了 Transformer 架构. 同时多模态预训练也有很多工作, VideoBERT, CBT 使用 BERT 同时学习 video-text pairs. ViLBERT, LXMERT 则使用了双流结构, Transformers 分别用于图像和文字, 在使用第三个 Transformer 来融合. 另一边 B2T2, VisualBERT, Unicoder-VL, VL-BERT 是单流结构. (Gan et al., 2020) 提出了使用多任务和对抗训练来提高性能. VALUE 使用了 probe tasks 来理解预训练模型.

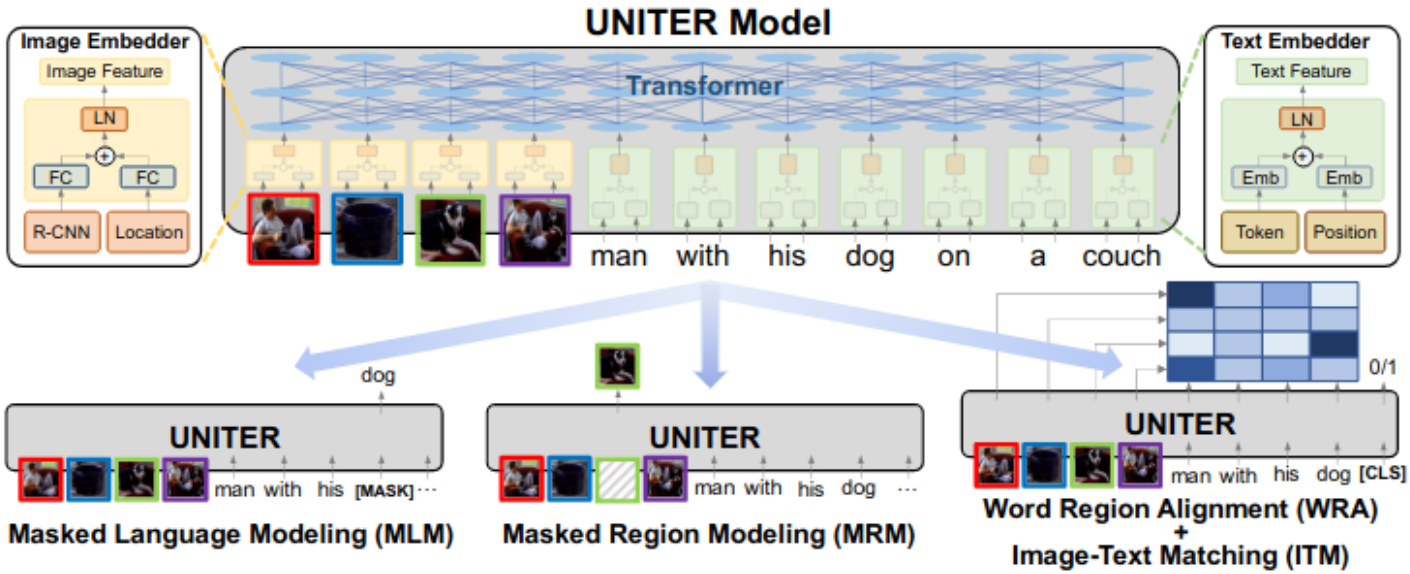
Remark 与其说它们 (UNITER 和 OSCAR) 提出了新的特征融合方法, 不如说它们提出了有效的基于 Transformer 的 V+L 预训练任务!

具体上, Image Embedder 使用 Faster RCNN 提取出的特征 (ROIPool 过), 以及区域位置特征送到两个 FC, 再相加, 再 LN. Text Embedder 类似 BERT, 将输入句子 token 成 WordPieces. 每个 subword 的 embedding 是词嵌入和位置嵌入加和, 再 LN.

UNITER

使用 Transformer, 启发于 BERT, UNITER 使用这些任务 pretrain:

1. Masked Language Modeling (MLM) conditioned on image
2. Masked Region Modeling (MRM) conditioned on Text
   - Masked Region Classification (MRC)
   - Masked Region Feature Regression (MRFR)
   - Masked Region Classification with KL-divergence (MRC-kl)
3. Image-Text Match-ing (ITM)
4. Word-Region Alignment (WRA, 使用 conditional masking 和基于最佳输运.



UNITER: Pretrain Tasks

Masked Language Modeling (MLM) 图片区域 $\{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$, 词 $\mathbf{w} = \{\mathbf{w}_1, \ldots, \mathbf{w}_T\}$, mask 指标 $\mathbf{m} \in \mathbb{N}^M$. MLM 中随机 mask 15% 的词, 使用 [MASK][2]. 目标是用周围的词预测 masked words

$$\mathcal{L}_{\mathrm{MLM}}(\theta) = -\mathbb{E}_{(\mathbf{w},\mathbf{v})\sim D} \log P_\theta \left(\mathbf{w}_{\mathbf{m}} \mid \mathbf{w}\backslash\mathbf{m}, \mathbf{v}\right) \tag{27}$$

Image-Text Matching (ITM) 加入了特殊 Token [CLS], 输入的是一些 regions 和词, 输出是否是匹配的文字/图像对. 具体的, 使用 [CLS] 对应的输出来作为 joint repr., 送到 FC+sigmoid 里的道 [0, 1] 的分数. 表示为 $s_\theta(\mathbf{w}, \mathbf{v})$. 使用二分类 CE

$$\mathcal{L}_{\mathrm{ITM}}(\theta) = -\mathbb{E}_{(\mathbf{w},\mathbf{v})\sim D} \left[y \log s_\theta(\mathbf{w}, \mathbf{v}) + (1-y) \log \left(1 - s_\theta(\mathbf{w}, \mathbf{v})\right)\right]\big) \tag{28}$$

训练中总是 50% 几率送进 paired/unpaired 的对.

Remark OSCAR 也同样进行了这两个任务.

[2]和 BERT 一致, 10% 变成随机词, 10% 不变, 80% 变成 [MASK]

UNITER: Pretrain tasks

Word-Region Alignment (WRA) 使用 transport plan$\mathbf{T} \in \mathbb{R}^{T \times K}$ 来表示 w/v 的对齐. 这是好的, 因为

- 自正则化, 元素和为 1
- 稀疏, 精确解必然只包含 2r-1 个非 0 元素
- 效率, 相比 linear solvers, 可以通过迭代法求解

具体的, 把 w,v 看作离散分布 $\boldsymbol{\mu}, \boldsymbol{\nu}$, 有 $\boldsymbol{\mu} = \sum_{i=1}^{T} \mathbf{a}_i \delta_{\mathbf{w}_i}$ and $\boldsymbol{\nu} = \sum_{j=1}^{K} \mathbf{b}_j \delta_{\mathbf{v}_j}$ 并且权值为单形上的顶点 $\mathbf{a} = \{\mathbf{a}_i\}_{i=1}^{T} \in \Delta_T$ and $\mathbf{b} = \{\mathbf{b}_j\}_{j=1}^{K} \in \Delta_K$, 那么定义两个离散分布之间的 OT 距离

$$\mathcal{L}_{\mathrm{WRA}}(\theta) = \mathcal{D}_{ot}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\mathbf{T} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i=1}^{T} \sum_{j=1}^{K} \mathbf{T}_{ij} \cdot c\left(\mathbf{w}_i, \mathbf{v}_j\right) \tag{29}$$

其中 c 是输运 cost, 实验中使用的是 consine unsim.(cos. dist.)

$$c\left(\mathbf{w}_i, \mathbf{v}_j\right) = 1 - \frac{\mathbf{w}_i^{\top} \mathbf{v}_j}{\left\|\mathbf{w}_i\right\|_2 \left\|\mathbf{v}_j\right\|_2} \tag{30}$$

T 的精确求解 intractable, 使用 IPOT 算法进行估计.

UNITER: Pretraing tasks

Masked Region Modeling (MRM) 类似 MLM, 15% 随机 drop feature. 训练来重建缺失的 visual feature, 但是由于这些特征是连续和高维的, 使用三个变种, 同一个 objective:

$$\mathcal{L}_{\mathrm{MRM}}(\theta) = \mathbb{E}_{(\mathbf{w},\mathbf{v})\sim D} f_\theta\left(\mathbf{v_m} \mid \mathbf{v}_{\backslash \mathbf{m}}, \mathbf{w}\right) \tag{31}$$

1. **Masked Region Feature Regression (MRFR)** 使用一个 FC 层把 Transformer 输出转换到输入同尺寸的空间上, 再使用 l2 回归

$$f_\theta\left(\mathbf{v_m} \mid \mathbf{v}\backslash\mathbf{m}, \mathbf{w}\right) = \sum_{i=1}^{M} \left\| h_\theta\left(\mathbf{v}_\mathbf{m}^{(i)}\right) - r\left(\mathbf{v}_\mathbf{m}^{(i)}\right) \right\|_2^2 \tag{32}$$

2. **Masked Region Classification (MRC)** 预测每个 masked 对象的 object semantic class. 使用 CE

$$f_\theta\left(\mathbf{v_m} \mid \mathbf{v}_{\backslash \mathbf{m}}, \mathbf{w}\right) = \sum_{i=1}^{M} \mathrm{CE}\left(c\left(\mathbf{v}_\mathbf{m}^{(i)}\right), g_\theta\left(\mathbf{v}_\mathbf{m}^{(i)}\right)\right) \tag{33}$$

3. **Masked Region Classification with KL-Divergence (MRC-kl)** 使用 soft-label, 即 object detector 的原始输出来做 CE/KL

$$f_\theta\left(\mathbf{v_m} \mid \mathbf{v}_{\backslash \mathbf{m}}, \mathbf{w}\right) = \sum_{i=1}^{M} D_{KL}\left(\tilde{c}\left(\mathbf{v}_\mathbf{m}^{(i)}\right) \| g_\theta\left(\mathbf{v}_\mathbf{m}^{(i)}\right)\right). \tag{34}$$

OSCAR

相比 UNITER 之类的类 BERT 工作, 额外使用 region feature 的 tag 信息.
输入为词-标签-图像 triplet$(\boldsymbol{w}, \boldsymbol{q}, \boldsymbol{v})$. 使用 pretrained-BERT 得到 $\boldsymbol{q}, \boldsymbol{w}$ 之间的对齐, 作为 OSCAR 的初始化; 被检测出 tag 的图像区域也有高的初始注意力权重.
具体上, $\boldsymbol{v}, \boldsymbol{q}$ 如下生成, 给定有 K 个对象区域的图片, Faster RCNN 用于提取图像特征 $(v', z), v \in \mathbb{R}^P$, 而 z 是一个 4/6 维向量 (左上/右下坐标 and/or 长宽), concat 二者. 使用线性映射来保证和词嵌入有相同的维度.
同时 Faster RCNN 用于检测一组高准确度的对象标签, $\boldsymbol{q}$ 是这些标签的词嵌入.

$$\boldsymbol{x} \triangleq [\underbrace{\boldsymbol{w}}_{\text{language}}, \underbrace{\boldsymbol{q}, \boldsymbol{v}}_{\text{image}}] = [\underbrace{\boldsymbol{w}, \boldsymbol{q}}_{\text{language}}, \underbrace{\boldsymbol{v}}_{\text{image}}] \triangleq \boldsymbol{x}' \tag{35}$$

用两种方式 interpret triplet.

OSCAR: Pretrain Tasks

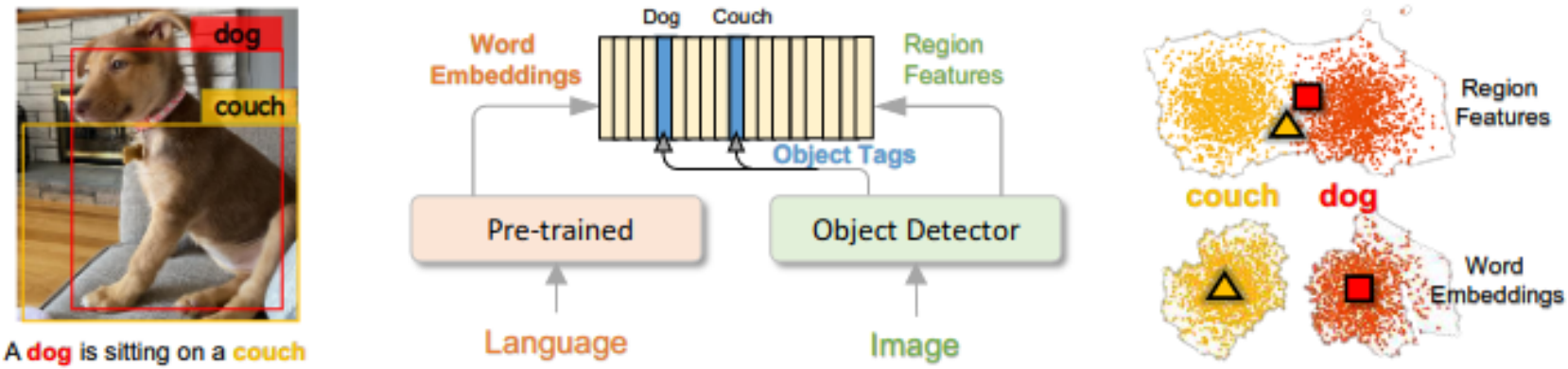
从字典角度看 $\mapsto$ Masked Token Loss(MTL). 定义 $\boldsymbol{h} \triangleq [\boldsymbol{w}, \boldsymbol{q}]$. 每次迭代, mask 掉 15% 的 token, 用特殊的 MASK token 代替. MTL 的目标时预测 masked token. 于是 MTL 为

$$\mathcal{L}_{\text{MTL}} = -\mathbb{E}_{(\boldsymbol{v},\boldsymbol{h})\sim\mathcal{D}} \log p\left(h_i \mid \boldsymbol{h}_{\setminus i}, \boldsymbol{v}\right) \tag{36}$$

这和 BERT 类似.
从模态角度看 $\mapsto$ Contrastive Loss. 考虑图像模态 $\boldsymbol{h}' \triangleq [\boldsymbol{q}, \boldsymbol{v}]$ 和语言模态 $\boldsymbol{w}$. 在 $\boldsymbol{q}$ 中 50% 堆积替换 tag. 在 encoder 输出后面加上一个 FC 层来预测 tag 是否包含了任何被替换的 tag(y=0), 或者没有变化 (y=0).
Reason(by authors): 够简单, 性能够好. 表现了两种 perspective.
Pretraining dataset: COCO, Conceptual Captions, SBU captions, flicker30k, GQA etc. 4.1M images, 6.5M triplets. OSCAR_B 使用 BERT base, AdamW, 1.0M steps, 5e-5 lr, BS 768.

OSCAR

tag embedding 作为 anchor 来分离一些相近的 semantic embedding.



(a) Image-text pair (b) Objects as anchor points (c) Semantics spaces

Fig. 2: Illustration on the process that OSCAR represents an image-text pair into semantic space via dictionary look up. (a) An example of input image-text pair (b) The object tags are used as anchor points to align image regions with word embeddings of pre-trained language models. (c) The word semantic space is more representative than image region features. In this example, dog and couch are similar in the visual feature space due to the overlap regions, but distinctive in the word embedding space.

OSCAR

VQA VQA 是多选回答. 使用 VQA 2.0, 基于 MSCOCO. 对于每个问题, 模型挑选 3129 个回答中的一个. 使用 [CLS] 的输出来作为特征, 并且送到线性分类器里. 视为 multi-label 分类问题, 给每个输出一个 soft target score, 取决于和 human answer 的相近度. 最后使用 CE, 使用预测分数和 soft-target 分数.

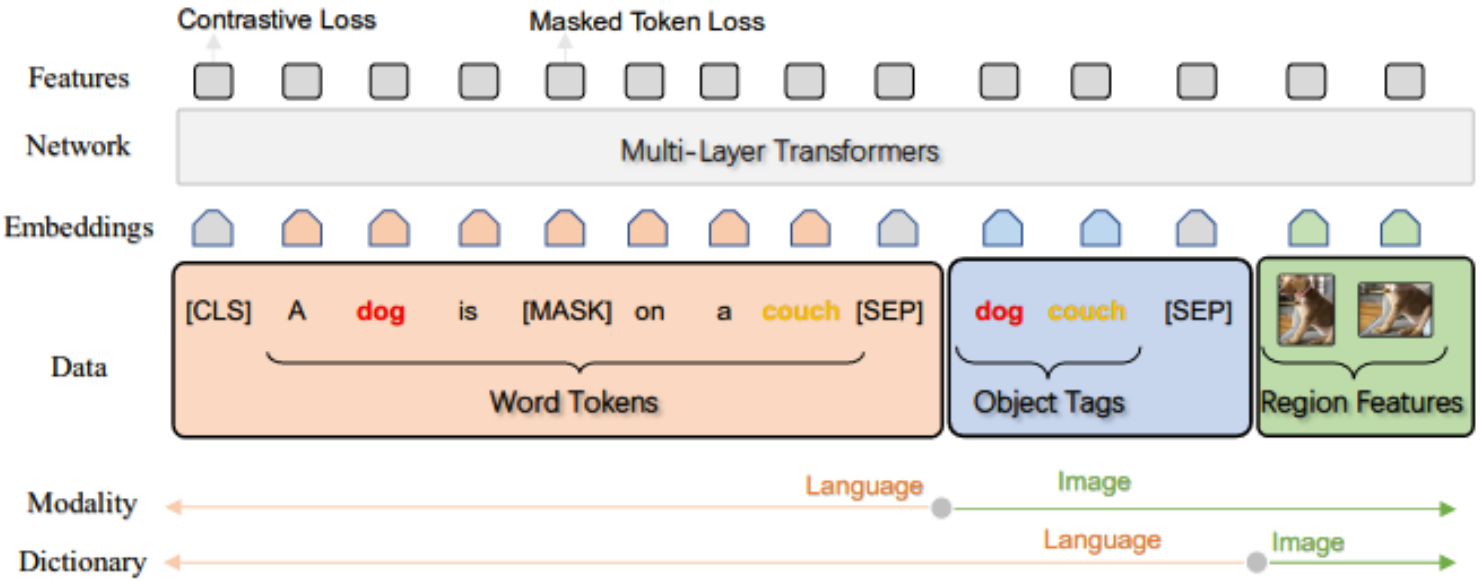


Fig. 3: Illustration of OSCAR. We represent the image-text pair as a triple [word tokens , object tags , region features], where the object tags (*e.g.*, "dog" or "couch") are proposed to align the cross-domain semantics; when