Motion-Appearance Co-Memory Networks for Video Question Answering

Abstract

比起$image\ QA$，$Video\ QA$有3个独特的属性：

1.它处理的是长的图片序列，且图片包含了丰富的信息，不论是在数量上还是在多元性上。

2.运动和外观信息通常彼此之间相互联系，且相互之间能够可能提供有用的关注线索。

3.不同问题需要不同数量的帧去推断出答案。

我们的网络是以$Dynamic\ Memory\ Network(DMN)$为基础，并且介绍了视频问答的新机制。

特别的是，我们的新机制有三个显著的方面

1.利用运动和外观线索产生注意力的一个联合记忆注意力机制。

2.一个时间卷积解卷网络用来产生多层上下文特征。

3.一个动态特征集成方法来构建不同问题动态的时间表征。

Introduction

DMN最开始是针对文本和图像问答提出的。它包含了一个记忆模块多周期的编码输入源，同时一个注意力机制在每一次循环利用读取过程关注到不同的内容上。

对于视频问答任务，我们提出了一个运动和外观联合记忆网络。我们的模型是以DMN为基础，所以我们和DMN共享了相同的术语，如特征、记忆、注意力。特别的是，我们通过双流模型将一个视频转化成一系列的运动和外观特征。运动和外观特征随后被喂入时间卷积和解卷积神经网络来建立多层上下文特征，该多层上下文特征有相同时间分辨率但是表征不同的上下文信息。这些上下文特征被用来作为记忆网络的输入特征。该联合记忆网络保存了两个分开的记忆状态，一个是运动，另一个是外观。为了共用模型和使运动和外观信息相互作用，我们设计了一个联合记忆注意力机制，该机制提取运动线索用于运动注意力的产生，提取外观线索用于运动注意力的产生。基于这些注意力，我们设计了动态特征集成方法，在每一轮特征编码时动态产生时间特征。

Related Work

前人的工作缺少运动分析和动态记忆更新机制。

为了正确回答基于视频的问题，视频的时间分析是必须的。

General Dynamic Memory Networks

因为我们的工作与DMN非常密切，我们首先介绍DMN的通用框架。它包含了4个明显的模块:输入模块、问题模块、事件记忆模块和回答模块。

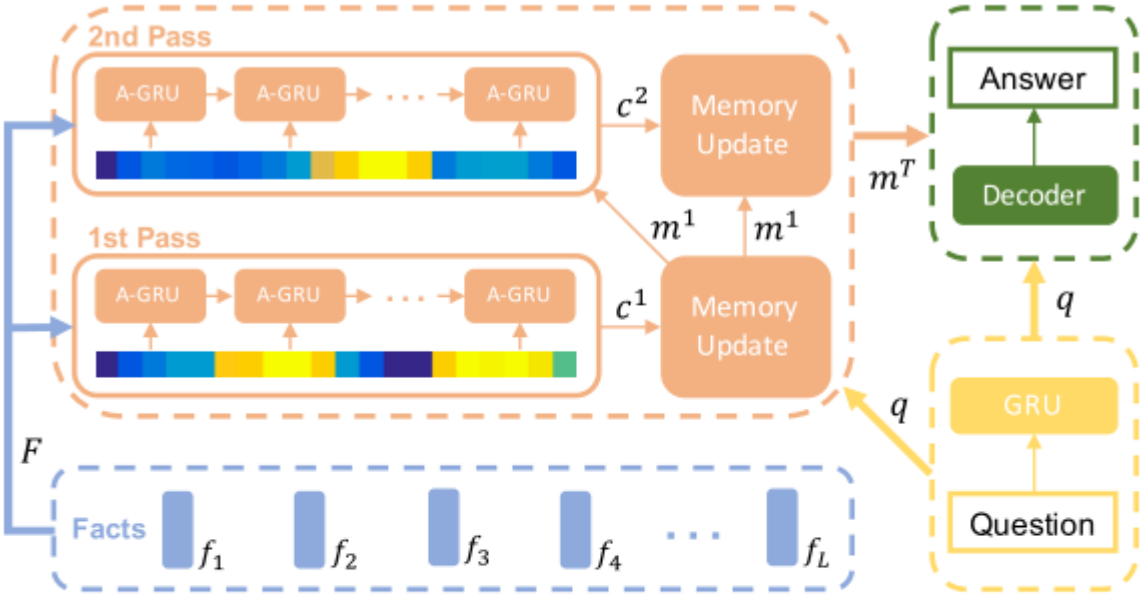

Figure 2. General Dynamic Memory Network (DMN) [20] architecture. The memory update process for the t-th cycle is : (1) the facts F are encoded by an attention-based GRU in episodic memory module, where the attention is generated by last memory m^{t-1}; (2) the final hidden state of the GRU is called contextual vector c^t, which is used to update the memory m^t together with question embedding q. The question answer is generated from the final memory state m^T.

Fact module

fact module将输入数据转化成一系列称为facts的向量，表示为$F = [f_1, f_2, \ldots, f_L]$，其中$L$是facts的数量。对于基于文本的问答，使用的是Gated Recurrent Unit(GRU)来编码所有文本信息；对于基于图像的问答，使用的是双向的GRU将局部区域的视觉特征编码为具有全局意识的facts。

Question module

问题模块将问题转化为嵌入q。使用GRU编码问题句子同时使用GRU最终隐藏状态做为问题嵌入。

Episodic memory module

事件记忆设计用来从facts中检索出相关信息。从facts抽取出与问题相关的信息更加有效，尤其是当能够从问题中获取传递推理时，事件记忆模块多个周期针对输入facts迭代反复，同时在每个周期之后更新记忆。在事件记忆模块中有两个机制：一个注意力机制和一个记忆更新机制。

假设在第t个周期，更新记忆为m^t，facts集合是$F = [f_1, f_2, \ldots, f_L]$，问题嵌入是$q$，那么注意力门$g_i^t$由下述公式给出：

$$g_i^t = F_a(f_i, m^{t-1}, q) \tag{1}$$

F_a是一个注意力函数，它将第i个fact向量f_i、第$t-1$周期的记忆m^{t-1}和问题q做为输入，并且输入一个标量g_i^t，这表征了第t周期的fact向量f_i的注意力值。

为了有效使用视频的顺序和位置信息，我们设计了一个基于GRU的注意力机制。在改进的GRU中，不在使用GRU中的原始更新门，将注意力门g_i^t用于修改后的更新公式：

$$h_i = g_i^t \circ \tilde{h_i} + (1 - g_t^t) \circ h_{i-1} \tag{2}$$

最终基于GRU的注意力隐藏状态用来做为上下文特征c^t，用于更新事件记忆m^t。c^t、问题嵌入q和周期$t-1$的记忆结合在一起，第t个周期的记忆更新公式为：

$$m^t = F_m(m^{t-1}, c^t, q) \tag{3}$$

F_m是记忆更新函数。最后的记忆m^T传递到回答模块，产生最终的答案。

Answer module

回答模块利用q和m^T来产生模型预测答案。不同的回答解码器可以应用到不同的任务中，如单个词答案用softmax输出层。

Motion-Appearance Co-Memory Networks

Multi-level Contextual Facts

将视频切成小的单元(一个序列帧)。对于每一个视频单元，我们使用双流卷积神经网络模型来提取出单元级的运动和外观特征。单元级的外观特征和运动特征序列分别表示为$\{a_i\}$和$\{b_i\}$。

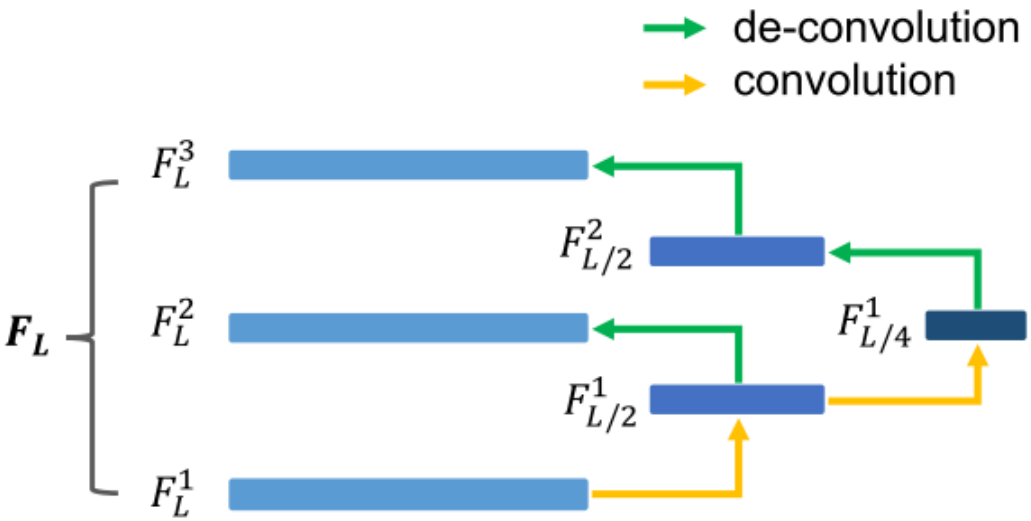

Figure 3. The input temporal representations are processed by temporal conv-deconv layers to build multi-layer contextual facts, which have the same temporal resolution but different contextual information.

建立多层的时间表征，每层表达不同的上下文信息，我们使用时间卷积层来建模时间上下文信息同时使用解卷积层来恢复时间分辨率，如图3。特别的是，最低层特征序列直接由单元特征构建，$A_L^1 = \{a_i\}$，$B_L^1 = \{b_i\}$。卷积层计算由几个尺度的时间特征序列组成的特征层次结构，扩展步长为2，F_L^1，$F_{L/2}^2$，$F_{L/4}^3$，...，如图三所示。要注意的是F可以是A(外观特征)或者是B(运动特征)。解卷积路径在时间上对特征序列进行粗糙的但是语义更强的上采样，得到更高分辨率特征F_L^2，F_L^3。因此，F_L^1、F_L^2和F_L^3有相同的分辨率但是有不同的时间上下文信息。图三上只有3层，更多的层可以通过增加更多的卷积和解卷积层来构建。$F_L = \{F_L^1, F_L^2, \ldots, F_L^N\}$称为上下文的facts。

Motion-appearance Co-Memory Module

Co-memory attention

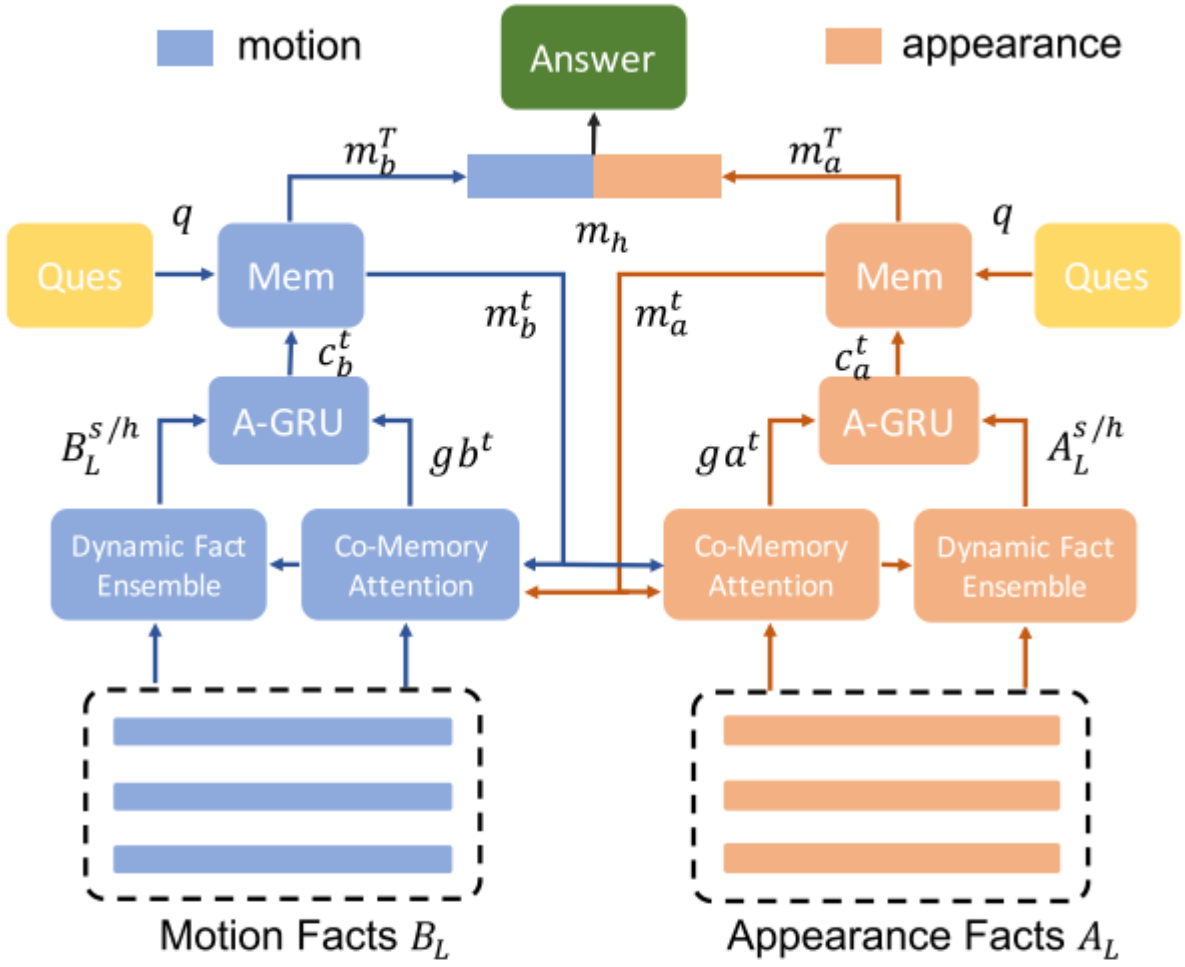

Figure 4. Co-memory attention module extracts useful cues from both appearance and motion memories to generate attention ga^t/gb^t for motion and appearance separately. Dynamic fact ensemble takes the multi-layer contextual facts $\mathbf{A}_L/\mathbf{B}_L$ and the attention scores ga^t/gb^t to construct proper facts $A_L^{s/h}/B_L^{s/h}$, which are encoded by an attention-based GRU. The final hidden state c_b^t/c_a^t of the GRU is used to update the memory m_b^t/m_a^t. The final output memory m_h is the concatenation of the motion and appearance memory, and it is used to generate answers.

Video QA的问题通常涉及到外观和运动。外观通常提供有用线索给运动注意力，换句话说引导运动要关注的内容，反之也一样。为了利用外观和运动的相互作用，我们设计了一个联合记忆注意力机制。特别的是，2个分开的记忆模块用来保存运动记忆m_b^t和外观记忆m_a^t。如前面所述，当网络读取运动facts来更新运动记忆，外观记忆提供有用线索来产生注意力；运动记忆对于更新外观注意力也是有帮助的。因此，在第t周期编码时，m_b^{t-1}和m_a^{t-1}用来产生运动和外观fact的注意力。当我们构建多个facts层级时，我们对每个fact向量的每个层级产生一个注意力分

数。对于fact的b_j^i来说，运动注意力门是$gb_{i,j}^t$，对于fact的a_j^i来说，外观注意力门是$ga_{i,j}^t$，其中t表示周期数，i表示fact表征的层级，j表示facts的第j步。

$$za_{i,j}^t = \tanh(\mathbf{W}_a^2(a_i^j + \mathbf{W}_a^1[m_a^{t-1}, q])) \tag{4}$$
$$ga_{i,j}^t = \mathbf{W}_a^4(za_{i,j}^t + \mathbf{W}_a^3[m_b^{t-1}, q])$$

$$zb_{i,j}^t = tanh(\mathbf{W}_b^2(b_i^j + \mathbf{W}_b^1[m_b^{t-1}, q])) \tag{5}$$
$$gb_{i,j}^t = \mathbf{W}_b^4(zb_{i,j}^t + \mathbf{W}_\mathbf{b}^\mathbf{3}[m_a^{t-1}, q])$$

$\mathbf{W}$是权重参数。$ga_{i,j}^t$和$gb_{i,j}^t$是在动态fact集成和记忆更新的注意力。

Dynamic fact ensemble

我们对外观和运动分别构建一个多层级的上下文facts集合$F_L = \{F_L^1, F_L^2, \ldots, F_L^N\}$，它们有相同的时间分辨率，但是表征不同的上下文信息。facts应该动态选择的原因有两个：(1)不同类型的问题可能获取不同层级的表征；(2)fact读取的多个周期期间，每个周期可能关注不同的层级的信息。我们设计了一个基于注意力的fact集成方法如图5所示。

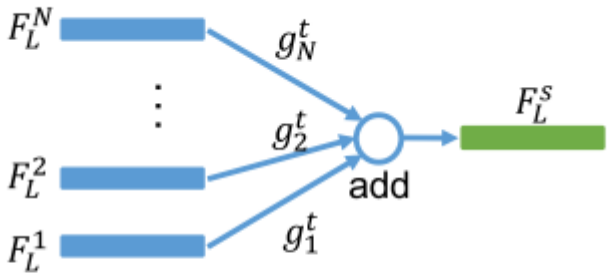

Figure 5. Multi-layer contextual facts are dynamically constructed via a soft attention fusion process, which computes a weighted average facts according to the attention.

为了简单起见，我们使用$g_{i,j}^t$来表示注意力门，实际上外观注意力门是$ga_{i,j}^t$，运动注意力门是$gb_{i,j}^t$。沿i轴，我们对$g_{i,j}^t$计算softmax来得到注意力分数$s_{i,j}^t$。

集成facts能够表示为：

$$F_t^s : \{f_j^t = \sum_{i=0}^{N} s_{i,j}^t f_j^i\}_{j=1}^L \tag{6}$$

$f_{i,j}$是F_L的第i层级第j步的fact向量。在fact编码处理后使用的注意力分数是由下面公式给出：

$$s_j^t = softmax(\frac{1}{N}\sum_{i=0}^{N} g_{i,j}^t), j = 1, 2, \ldots, L \tag{7}$$

这里的softmax是沿着j轴计算的。

Memory update

fact编码过程分别对运动和外观执行，采用了一个基于注意力的GRU在第t个周期对于外观和运动分别产生上下文向量c_a^t和c_b^t。运动记忆m_b^t和外观记忆m_b^t分别用以下公式更新：

$$m_a^t = \mathrm{FC}([m_a^{t-1}, q, c_a^t]) \tag{8}$$

$$m_b^t = \mathrm{FC}([m^{t-1}, q, c_b^t]) \tag{9}$$

FC是全连接层，ReLU用来做非线性激活函数。最后输出记忆m_h是m_a^T和m_b^T的串联。

Answer Module

接着TGIF-QA，我们将TGIF-QA的4个任务构建成3个不同类型：多项选择，开发性数字和开放性词汇。

对于多项选择，我们对记忆状态m_h使用线性回归函数，同时输出每个候选回答的实值分数。

$$s = \mathrm{W}_m^T m_h \tag{10}$$

这个模型在正确回答s_p的分数和错误回答s_n的分数之间使用hinge loss，$\max(0, 1 + s_n - s_p)$是最佳的。这个解码器用于解决重复行为和状态转换任务。

对于开放性数字，我们也对记忆状态m_h使用线性回归函数，同时输出是一个整数值回答。

$$s = [\mathrm{W}_n^T m_h + b] \tag{11}$$

[.]意味着取整数。我们在真值和预测值之间采用ℓ_2 loss来训练模型，这用来解决重复数字任务。

对于开放性词汇，我们将这做为一个分类问题。我们对记忆状态m_h使用线性函数，后面采用softmax层产生答案。

$$\mathbf{o} = \mathrm{softmax}(\mathbf{W}_w^\top m_h + \mathbf{b}) \tag{12}$$

交叉熵loss用来训练模型，这个解码器用于Frame QA任务。

Evaluation

TGIF-QA dataset:该数据集由来自71K个动画 $\mathrm{Tumblr\ GIFs}$的165K对问答组成。共有4个任务:重复次数计数、重复行为、状态转换和画面问答。

Implementation Details

Appearance and motion features

由于TGIF-QA的每秒帧数(FPS)不同，我们使用相应GIF文件的FPS从所有GIF中提取帧。长的视频被切割成小的单元，每个单元包含6帧。

为了提取单元级视频特征，我们使用$\mathrm{ResNet\text{-}152}$来处理一个单元的中心帧并且将$\mathrm{ResNet\text{-}152}$的pool5层的输出($\in \mathbb{R}^{2,048}$)做为外观特征。为了利用运动信息，我们抽取视频单元里面的光流，并且使用双流模型的流CNN来得到单元级的流特征。具体的来说，将6个连续帧的单元送入预训练流CNN模型(BN-Inception network)来计算相邻帧的双向密集光流。随后，我们得到global pool层($\in \mathbb{R}^{1,024}$)的特征映射做为初始光流特征。最后，我们通过平均池化下采样特征维度并得到2048维向量做为我们双向光流特征。在这个过程中，如果每个单元中间没有足够的帧，我们会填补第一帧或最后一帧。我们设置视频特征的时间分辨率为34，长的特征序列被切割，短的被填充。

Contextual facts

卷积-解卷积网络的每一层输出通道是1024，时间卷积核的尺寸为3，步长为1，解卷积层的步长为2，最大池化的尺寸为2，步长为2.我们构建了上下文facts共3层。

Co-memory module

记忆状态m_a和m_b的尺寸设置成1024。用于fact编码的GRU隐藏状态的尺寸为512。$za_{i,j}^t$和$zb_{i,j}^t$是512维。

Question and answer embedding

对于问题的每个词，我们使用预训练的词嵌入模型转化为300维的向量。问题的所有词通过两层的GRU(隐藏状态的尺寸为512)进行处理，最终的隐藏状态做为问题嵌入。对于行为转换和重复行为任务来说，候选答案是一个词汇序列，于是，我们使用相同的方法将答案编码。

Training details

我们设置batch size为64。Adam优化器用来优化模型，学习率设置为0.001。每个任务训练50轮。