RobustScanner: Dynamically Enhancing Positional Clues for Robust Text Recognition

Paper: 2020, 商汤科技

源码地址：无

摘要

基于Attention的编解码框架对于缺少上下文的文本（随机字符序列）效果较差。本文探究了解码过程发现，解码器解码时不仅使用文本信息，同时还使用位置信息。（现存的方法过度依赖文本信息，导致注意力漂移问题）。

本文提出一个模型，包括一个位置感知模块，使得编码输出的特征包括自身位置信息；一个Attention模块，一个动态融合模块，通过逐元素的门控机制构建更robust的特征。

相关工作

SAR-2019：构建2D的Attention机制

DAN-2020：将误识别归因于注意力漂移，方法是将注意力机制同历史的预测结果解耦；本文则是提出一个位置增强分支，动态的调整解码时文本信息和位置信息的比例。

Self-Attention-2018：位置信息模型

问题讨论

Attention解码过程回顾：

- 解码的LSTM接受字符和上一步的隐态，得到当前步的隐态ht；
- 将ht向量作为attention模块的query向量，构建特征的权重图；
- 聚合卷积特征图F和权重图，得到glimpse向量gt；（**有的方法进一步将ht拼接上去构成新gt**）（**另外1D的Attention没有直接利用特征图F，而是用ht做Attention**）
- 再加线性层和softmax将gt向量映射到分类空间，输出预测结果；

$$\mathbf{h_t} = \mathrm{LSTM}(\mathbf{x_t}, \mathbf{h_{t-1}}), \mathbf{x_t} = \begin{cases} \mathbf{y_{t-1}} & \text{if } t > 1 \\ <\text{start}> & \text{if } t = 1 \end{cases},$$

$$\alpha_{ij}^t = \mathrm{softmax}(\mathbf{h_t}^T \mathbf{f_{i,j}}),$$

$$\mathbf{g_t} = \sum_{ij} \alpha_{ij}^t \mathbf{f_{i,j}}.$$

$$\mathbf{y_t} = \mathrm{softmax}(\mathbf{W}\mathbf{g_t} + b),$$

Attention解码过程的讨论：

从上面公式可知，给定特征图F和网络参数，预测的结果仅依赖于query向量ht。那么query向量ht编码了什么信息，使得与之相关的注意力权重能够聚焦，并使得分类器可以正确的识别序列中的某个字符？

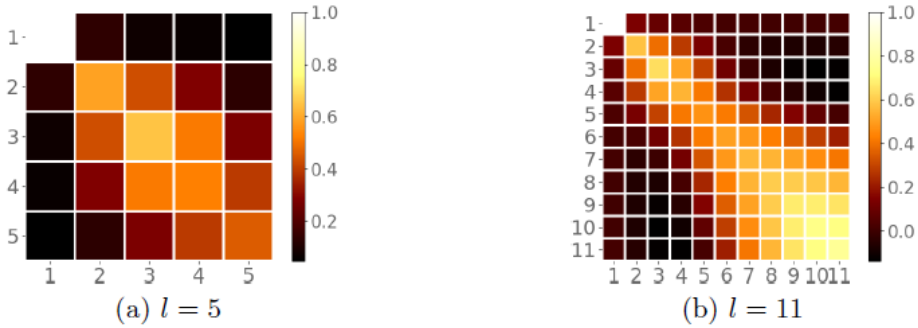
> 作者观察到对于不同的文本序列，其第一步的query向量h1保持不变（并没有上下文信息），而第一个字符依然能被正确识别，侧面说明query向量包含位置信息（例如字符的位置序号等）？？

作者分析了在相同时间步下，不同文本序列的query向量的相似度，进一步验证ht编码包含位置信息

- 构建等长的文本序列集合，计算不同文本下，各个时间步ht（等于位置）的平均余弦相似度

$$S_l(i,j) = \frac{\sum_{m\neq n}^{|\widehat{\mathcal{I}}_l|} \cos(\mathbf{h_i^{m,l}}, \mathbf{h_j^{n,l}})}{\left|\widehat{\mathcal{I}}_l\right| (\left|\widehat{\mathcal{I}}_l\right| - 1)},$$

- **可视化**l=5和l=11的结果：
  - 相同位置的query相似度明显高于不同位置
  - 随着时间步增加，相同位置和相邻位置相似度对比度变小（变得模糊）



(a) $l = 5$ (b) $l = 11$

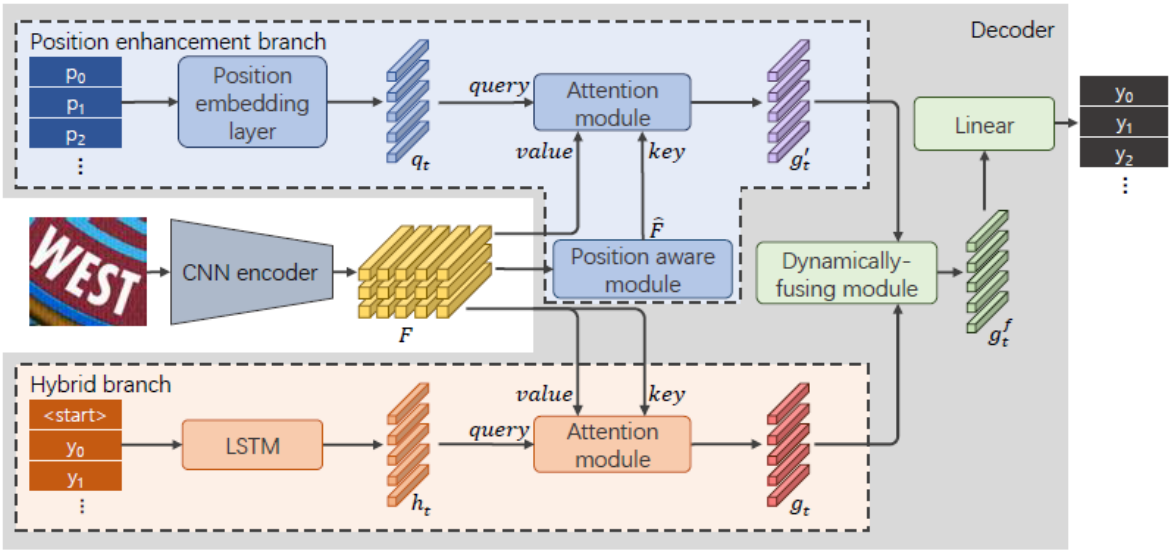
> 第i个字符其实是改变的（如果没有位置信息，那么字符不同，相同位置的相似度不应该很高）

讨论结论：

- qurey向量包含上下文和位置的混合信息
- 解码时随着时间步增加，文本信息变强同时位置信息减弱

模型设计细节

整个模型有四个部分包括CNN，Hybrid分支，位置增强分支和动态融合模块



CNN提取特征：

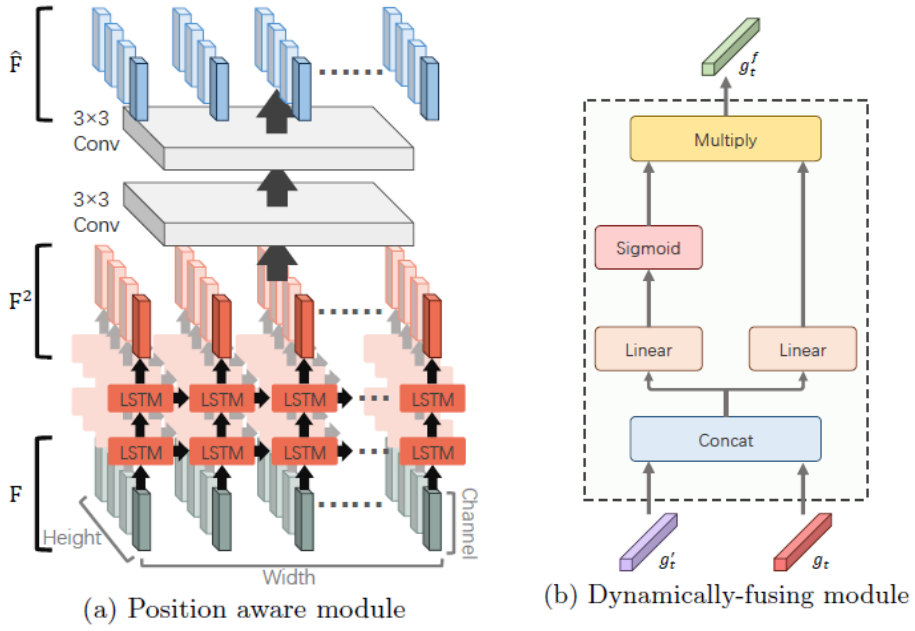
- 31层ResNet和SAR-2019一致

Hybrid分支：

- **两层**LSTM解码器，包含128个隐含单元；接受自身隐含态和预测的字符作为输入，产生qurey向量ht；
- 基于特征图F和ht做Attention，得到glimpse向量gt （参考SAR-2019）

位置增强分支：[嵌入层编码，如何保证感知位置的准确？]

- 位置嵌入层：对解码的时间步进行编码，每个时间步输入one-hot向量输出具体的embedding向量qt（embedding向量在不同的解码序列中保持不变？？）
- 位置感知层：两层LSTM(128)，遍历特征图所有行；再用两个3x3卷积+中间一个ReLu提取特征向量（2D）；



(a) Position aware module (b) Dynamically-fusing module

- Attention模块：以嵌入层输出qt作为query向量，同位置感知输出的特征图做Attention得到glimpse；
- 动态融合模块：拼接两个向量，用两个线性层（其中一个有softmax）构建自注意的结构，得到最终输出；

其他细节

数据集：用MJSynth和SynthText训练，在IIIT，SVT，IC13， IC15，SVTP，CUTE80上测试

训练配置：

- Adam优化器，lr_init=0.001, 包含5个epochs，第3代lr=0.0001， 第4代lr=0.00001
- batch=128（4块12G的Titan X）；图片高度设置为48，宽度保持比例（同SAR-2019一致）
- query向量维度=128（减少计算量）；最大的位置嵌入数量=36

实验结果

各benchmarks下的精度：

- SVT效果明显比SAR方法差
-

Method	Training Data	Regular Text			Irregular Text		
		IIIT5K	SVT	ICDAR 2013	ICDAR 2015	SVTP	CUTE 80
Cheng *et al* [7]	MJ + ST	87.4	85.9	93.3	70.6	-	-
Cheng *et al* [8]	MJ + ST	87.0	82.8	-	68.2	73.0	76.8
Shi *et al* [42]	MJ + ST	93.4	**93.6**	91.8	76.1	78.5	79.5
Zhan and Lu [57]	MJ + ST	93.3	90.2	91.3	76.9	79.6	83.3
Gao *et al* [10]	MJ + ST	94.0	88.6	93.2	**77.1**	80.6	88.5
Bai *et al* [3]	MJ + ST	88.3	87.5	94.4	73.9	-	-
Luo *et al* [29]	MJ + ST	91.2	88.3	92.4	68.8	76.1	77.4
Wang *et al* [50]	MJ + ST	93.3	88.1	91.3	74.0	80.2	85.1
Lyu *et al* [31]	MJ + ST	94.0	90.1	92.7	76.3	82.3	86.8
Xie *et al* [52]	MJ + ST	82.3	82.6	89.7	68.9	70.1	82.6
DAN [51]	MJ + ST	94.3	89.2	93.9	74.5	80.0	84.4
Bartz *et al* [4]	MJ + ST	94.6	89.2	93.1	74.2	**83.1**	89.6
Bleeker *et al* [6]	MJ + ST	94.7	89.0	93.4	75.7	80.6	82.5
Long *et al* [28]	MJ + ST	93.7	88.9	92.4	76.6	78.8	86.8
Baek *et al* [1]	MJ + ST	87.9	87.5	92.3	71.8	79.2	74.0
RobustScanner	MJ + ST	**95.3**	88.1	**94.8**	**77.1**	79.5	**90.3**
SAR [25]	MJ + ST + R	95.0	**91.2**	94.0	78.8	**86.4**	89.6
RobustScanner	MJ + ST + R	**95.4**	89.3	**94.1**	**79.2**	82.9	**92.4**