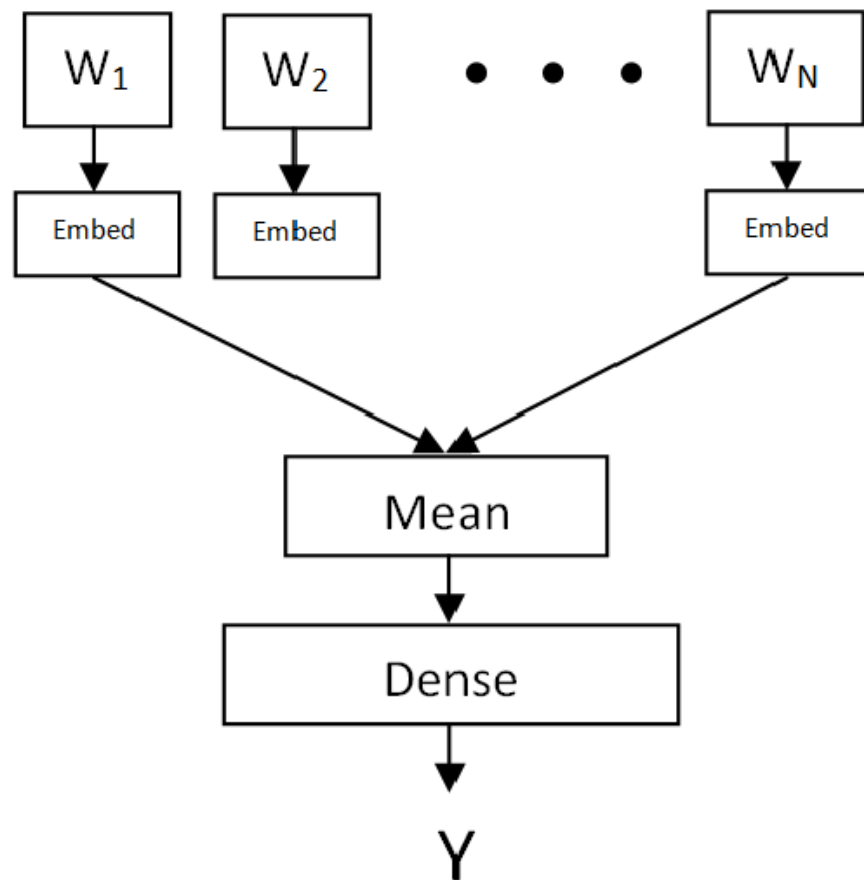


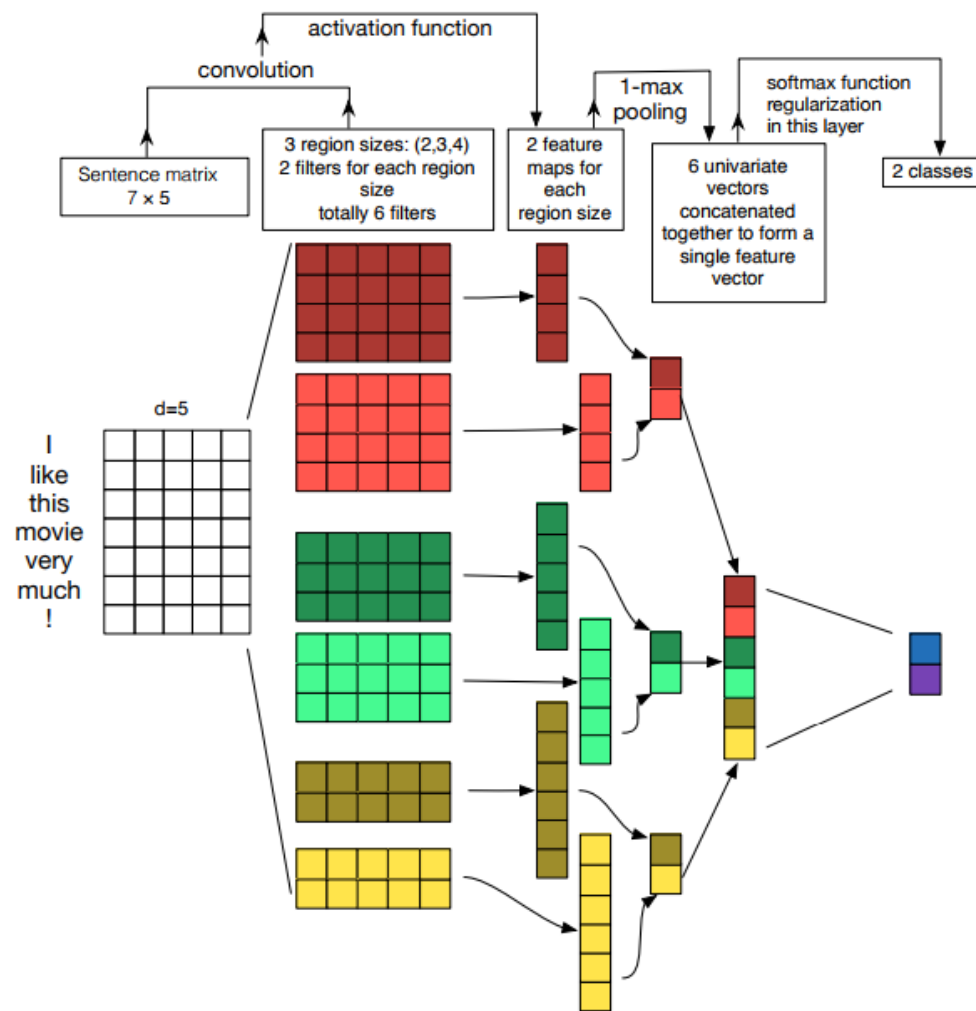
# fastText

- 取所有词向量的均值作为文本的特征表示;
- 将N-gram融入特征中,  
E.g. “文本分类”  
Unigram: “文”, “本”, “分”, “类”;  
Bigram: “文本”, “本分”, “分类”;  
Trigram: “文本分”, “本分类”.
- N-gram组合过多? Hash!
- 简单有效, 也是训练词向量的方法之一。



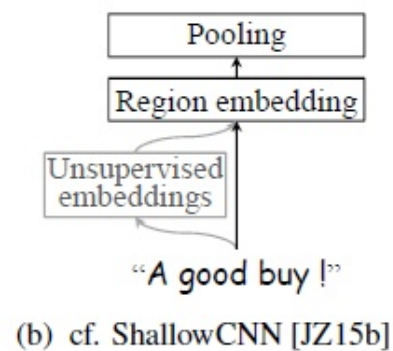
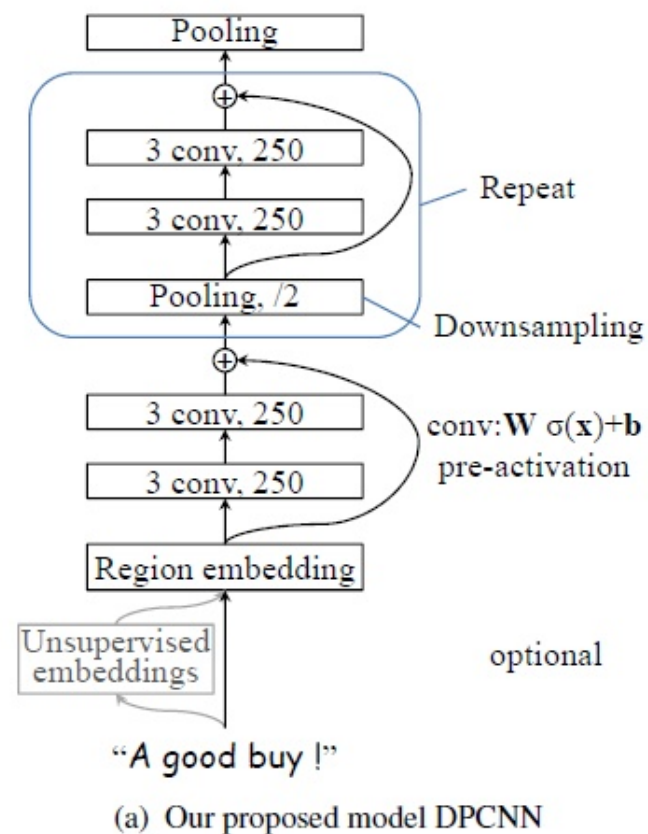
# TextCNN

- 假设词向量是 $K$ 维的,  
用 $N \times K$ 的卷积核作为 $N$ -gram提取器;
- 沿着卷积核的输出做max pooling;
- 用dropout防止全连接层过拟合;
- 用预训练的词向量进行初始化,  
在训练中进行微调。
- 卷积层能有效提取局部语义信息,  
但不擅长提取全局特征。



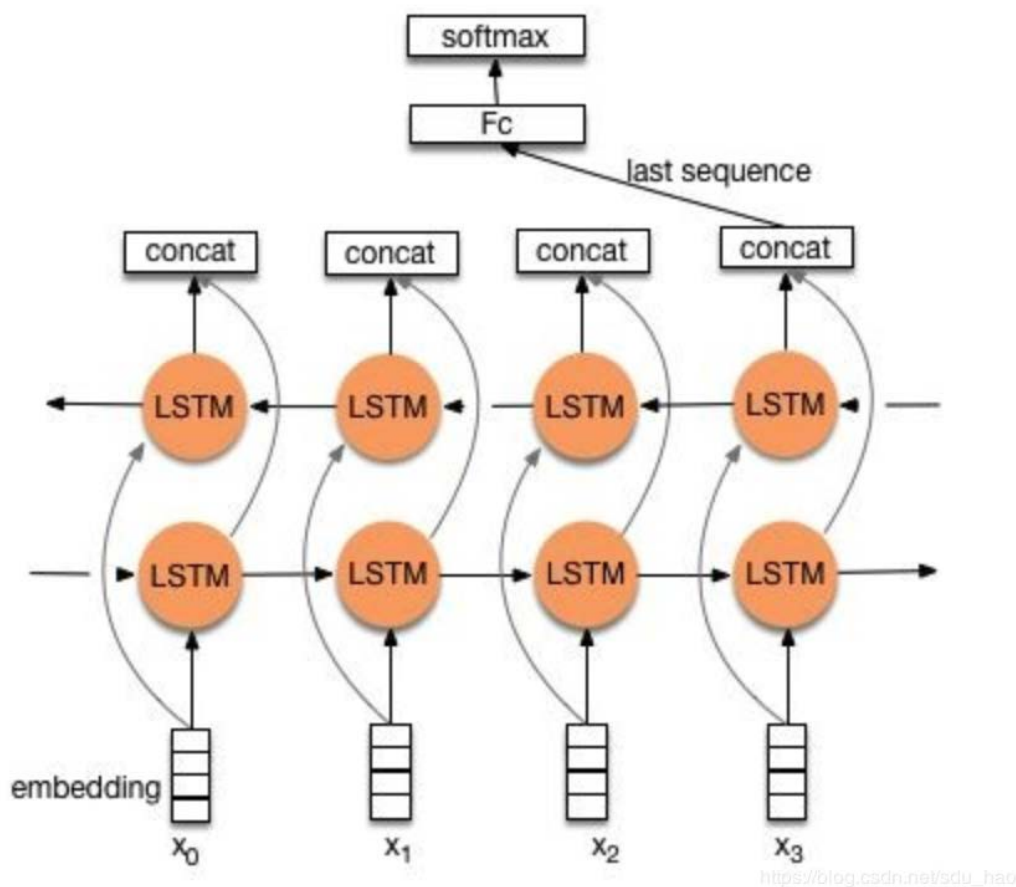
# Deep Pyramid CNN

- 在TextCNN的基础上，加入了重复多次的池化-卷积-卷积操作；
- 每经过一次池化，序列的长度就缩短一半，
- 越靠上的卷积层越能提取出宏观层面的信息；
- 因为序列长度不断减半，计算资源的消耗得到了有效的降低。
- 将卷积层的输入和输出加在一起  
(skip connect from ResNet),  
使深层网络的训练更有效。



# BiLSTM

- 取双向LSTM最后一个时间步上的隐状态作为文本的特征表示。
- RNN擅长捕捉长距离关系，但不擅长提取局部语义信息。
- 不适用于长文本分类。

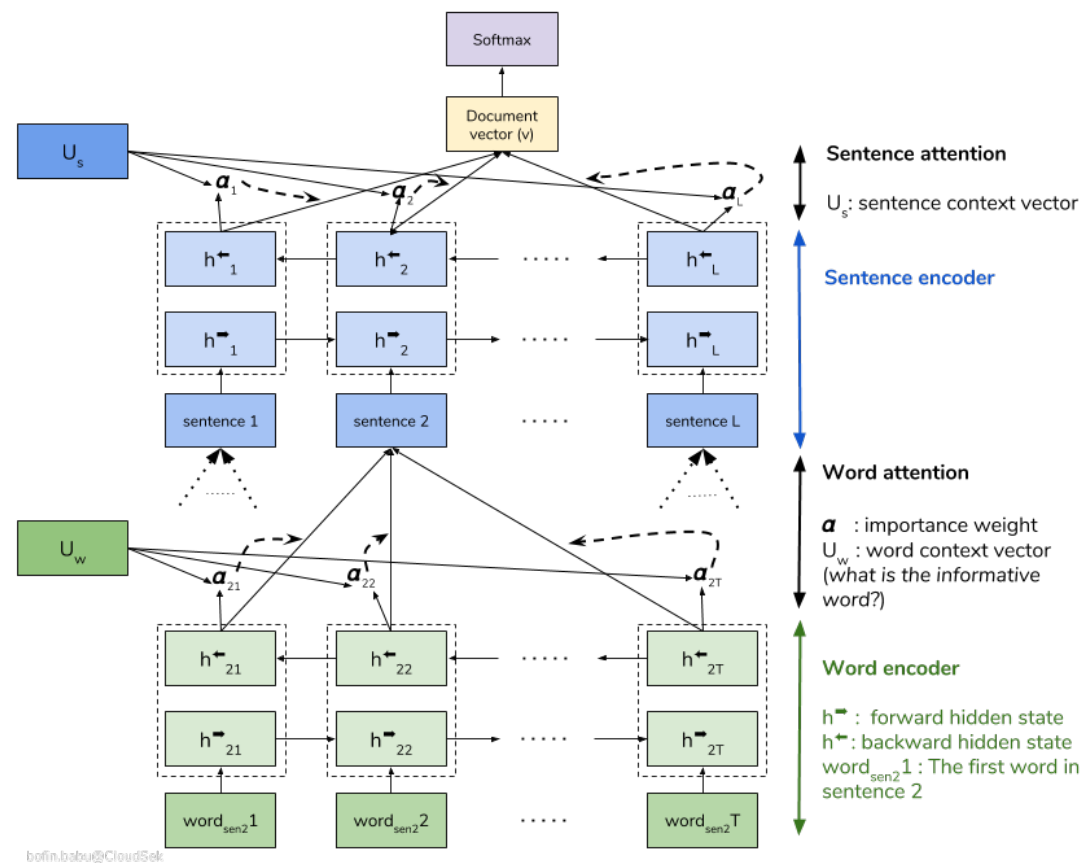


# BiLSTM with Attention

- 取双向LSTM所有时间步的隐状态输入Attention层,
- 取Attention层的输出作为文本的特征表示。
- Attention机制:

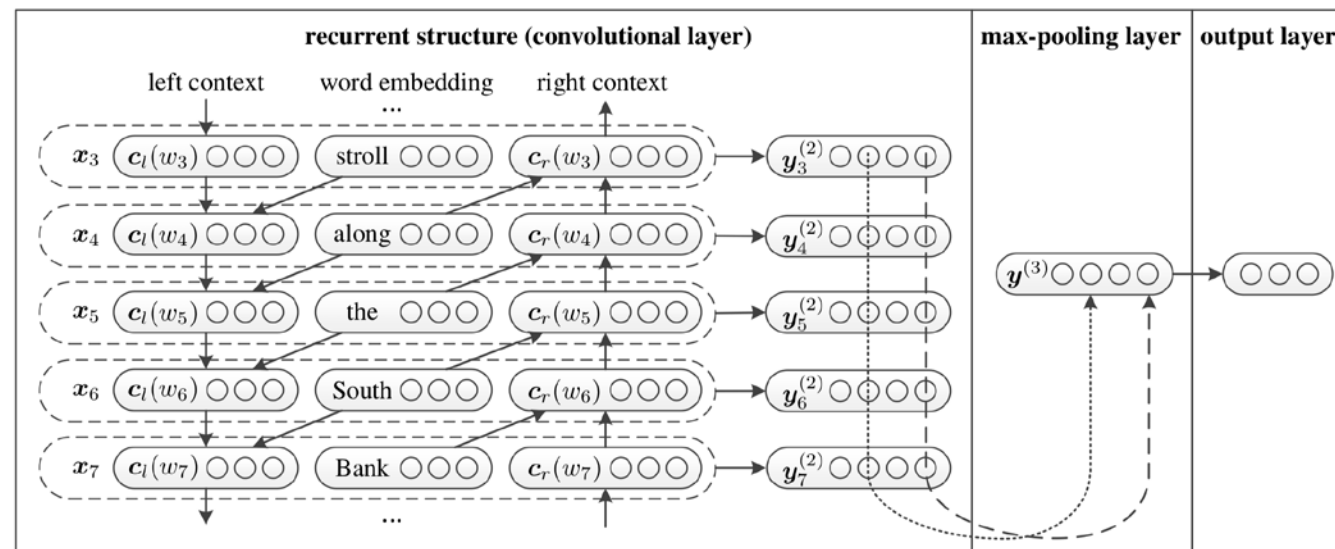
$$a_i = \text{softmax}(f(h_i))$$
$$f(h_i) = v_i^T \text{ReLU}(Wh_i + b)$$

- Attention机制增强了模型提取全局信息的能力, 但损失了语序关系。



# TextRCNN

- 将BiLSTM的输入和输出拼接在一起, 经过激活函数后做max pooling.
- 将TextCNN中的卷积层换成了RNN(LSTM).
- 用RNN做文本分类的又一种策略。



# BERT

- 基于Transformer的大型预训练网络, 性能state-of-the-art, 仍然是研究的热点。
- Whole word Mask.
- 在训练数据集上对其进行微调。

