NLP1 Report

潘宜城

2019 年 10 月 27 日

目录

1 问题描述

等待完成

2 解决方案

从标题到故事线再到故事，前一步采用已经获得的故事线作为数据。

3 Language Model

采用语言模型的思路解决该问题。此方法先从数据中学习语言模型，也就是根据输入的词序列输出词序列的方式，然后将测试数据结合故事线输入得到测试输出。

3.1 Data Representation

本节介绍数据在模型内的表达方式，也即是模型训练的输入格式。

ROC story是由一对对的标题故事组成的数据，这里不区分它们的差别，都将其当作一段长文本的一部分，组成一个表示整个训练数据的词序列。取一组训练数据时，假设从单词w_i开始，长度为s，那么训练数据的输入为$w_i, ..., w_{i+s-1}$，输出为$w_{i+1}, ..., w_{i+s}$，即下一个词。

生成测试输出时，则先按照测试样本的标题和故事线词依次输入，然后取模型的输出，并且当作之后的输入，直到输出终止符号。

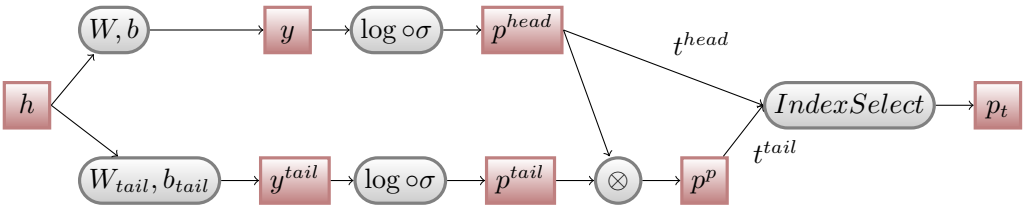
3.2 Model

此处画模型展开示意图。

3.3 SplitCrossEntropy

考虑到不同的词的频率不同，而在使用CrossEntropy作为损失函数的时候所有的词都将具有相同的重要性，这里根据词的频率将其划分到不同的区间，然后计算划分到较低频率区间的词的概率的时候加上对其概率的频率修正，即坟墓概率。概率修正的参数作为损失函数的可学习参数，与模型一起学习。

该Loss对不同区间的单词赋予不同的权重，然后再进行联合概率的计算。简单地看，对一个输出词向量，损失的计算如图所示。



3.3.1 相关符号

n_h隐藏层大小，也是循环网络输出的维度。

n_v词库的大小，即词的总数量，也是下面的划分的右边界。

$S = \{s_0, s_1, ..., s_{m-1}, s_m\}$用于将词分为不同区间的分割点，以词的字典索引为准。划分区间个数为m. 为了包含整个词库，其中s_0为0,s_m为词库大小。

$W_{tail} \in R^{(m-1)\times n_h}, b_{tail} \in R^{m-1}$ is the weight and bias that used to calculate the probability of the tombstones.

3.3.2 Forward process

这里根据隐藏层输出H,目标词序列t，输出层权重$W \in R^{n_v \times n_h}, b \in R^{n_v}$进行计算。

首先将H变形为$R^{n \times n_h}$，然后根据目标词的索引t的位置和分组参数$S = \{s_0, s_1, ..., s_{m-1}, s_m\}$进行分组。结果表示为$\tilde{H} = \{\tilde{H}_1, ..., \tilde{H}_m\}, \tilde{t} = \{\tilde{t}^1, ..., \tilde{t}^m\}$。例如，$\tilde{t}^i$包含了所有输出索引中索引在范围$[s_{i-1}, s_i)$内的索引，$\tilde{H}$也是同样分组的。每个分组内词的数量记为$n_i$。

然后计算头部区域$[s_0, s_1)$的头部词概率以及坟墓的概率。这个过程使用的权重是

$$W_{head} = \left(\begin{array}{c} W[s_0, :] \\ W[s_0 + 1, :] \\ \vdots \\ W[s_1 - 1, :] \\ \hline W_{tail} \end{array}\right), b_{head} = \left(b_{s_0} b_{s_0+1} \cdots b_{s_1-1} \middle| b_{tail}\right)$$

同理使用$\tilde{H}$在行上连接，得到

$$\tilde{H}' = \begin{pmatrix} \tilde{H}_1 \\ \tilde{H}_2 \\ \vdots \\ \tilde{H}_m \end{pmatrix}$$

然后根据上述参数得到结果

$$Y^{head} = \tilde{H}' \times W_{head}^T + b_{head}$$

再对Y使用对数Softmax得到对应的概率。

$$P^{head} = \log\left(Softmax(Y^{head})\right)$$

对于属于头部词，他们的交叉熵直接为对应的项的和，如下:

$$E_{head} = -\sum_{i \in [0,\, n_1)} P^{head}_{i,\tilde{t}_i^1}$$

而对于其他分组的词，概率需要包括坟墓的概率。假设我们考虑分组$p(p \in [2, m])$，它所包含的词索引的范围是$[s_{p-1}, s_p)$，包含的词在坟墓概率中的索引在$[i_p, i_p + n_p)$。我们使用输出层和这些索引对应的权重$W[s_{p-1} : s_p, :], b[s_{p-1} : s_p]$计算词的分布概率。

$$\begin{aligned} P^{p\prime} &= \log\left(Prob(Tomb_p) \times Prob(Words|Tomb_p)\right) \\ &= \left(\log\left(Prob(Tomb_p)\right) + \log\left(Prob(Words|Tomb_p)\right)\right) \\ &= P^{head}[i_p : i_p + n_p, -(m + 1 - p)] + P^p \end{aligned}$$

为了计算上面的概率, 假设Y^p表示分组p的输出值。因而就有

$$\begin{aligned} Y^p &= \tilde{H}_p \times W[s_{p-1} : s_p, :]^T + b[s_{p-1} : s_p] \\ P^p &= \log\Big(Softmax(Y^p)\Big) \end{aligned}$$

所以分组p的交叉熵为

$$E^p = -\sum_{i \in [0, n_p)} P^{p\prime}_{i,\tilde{t}_i^p}$$

综上，总交叉熵H等于$\frac{1}{n} \times \left(E^{head} + \sum_p E^p\right)$.

3.3.3 Log probability

这一步是用来计算任意的隐藏层输出最后的输出词概率分布的。和上面的计算过程类似，如果输入形状为$(k \times n_h)$，那么输出为$(k \times n_v)$, 每行都是输出词的概率分布。

3.4 Weight Dropout

Weight Dropout是用在循环神经网络内部的正则方式，通过随机去除循环神经元中隐藏层到隐藏层的连接，防止循环神经网络的过拟合。

3.5 Embedding Dropout

Embedding Dropout是嵌入层采用的正则项。通过随机去除输入序列中某些词的嵌入输出，即消除这个词，进行正则。

3.6 Locked Dropout

Locked Dropout是对序列的向量采用相同的Dropout的正则方式。考虑到序列中每个向量之后都会经过相同的运算，采取保持一致的Dropout有利于模型利用相同的信息。

3.7 Experiments

本节介绍采用的实验的参数和结果。

Model	Em size	Hidden size	Test ppl	Bleu 4
model tied with wdrop	800	800	22.85	0.4074
model tied without wdrop	800	800	27.22	0.3908
model tied with wdrop	800	800	35.28	None
model tied data fix	400	400	None	None

4 Attention Decoder Model

参考NLP FROM SCRATCH: TRANSLATION WITH A SEQUENCE TO SEQUENCE NETWORK AND ATTENTION实现Decoder中的注意力

机制。这里的数据模型是序列到序列，通过限制序列的长度为定长，则可以使用注意力层对输入序列的输出进行加权，然后再依此生成目标的序列。

4.1 Corpus Data Preprocess

本节介绍对ROC Stories数据预处理成适合模型训练的过程。

由于注意力机制是针对序列的词施加注意力权重，这里的训练数据应该以整个序列的方式输入，同时输出也应该是同样的整个序列。考虑到ROC Stories的数据特征，这里将生成好的故事线作为输入，故事作为输出。为了使用批量训练加速训练过程，需要对不同长度的输入数据进行长度对齐操作，输出也是。若用S_{max}表示最长序列长度，N表示批量的个数，x_i^b, y_i^b表示$Batch\ b$中输入和输出序列的第i个词，那么对齐处理后的数据会具有如下矩阵所示的形状。

$$X = \begin{bmatrix} x_0^0 & x_0^1 & \cdots & x_0^{N-1} \\ x_1^0 & x_1^1 & \cdots & x_1^{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} Y = \begin{bmatrix} y_0^0 & y_0^1 & \cdots & y_0^{N-1} \\ y_1^0 & y_1^1 & \cdots & y_1^{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

4.2 Model