Report

王天乐

Data Preprocessing

由于数据是tsv格式，且列之间采用Tab分割，所以利用python的csv库可以比较容易的进行读取:

```
reader = csv.reader(open("test.tsv", "r", encoding="utf-8"), delimiter='\t')
```

由于考虑到是文本匹配任务，所以文本次序并不重要，对于train data我们可以把label不变的情况下，交换一下左右的次序，这样可以把样本量翻倍，有助于训练。这样样本数目其实已经足够大了(70w+)，我尝试过进一步拓展增强数据，通过随机采样得到负样本，通过重复得到正样本，但测试发现相比于时间上的消耗这样带来的受益相当有限。

Model Introduction

我总共跑了两类模型，探究了目前SOTA的交互式匹配模型和特征式匹配模型的表现。

模型分别是交互式的BERT、RoBERTa和特征式的CoSENT，下面我将简单介绍一下这两类模型：

BERT[1] 是一个强大的预训练语言表征模型。它强调了不再像以往一样采用传统的单向语言模型或者把两个单向语言模型进行浅层拼接的方法进行预训练，而是采用新的Masked Language Model (MLM)，以致能生成深度的双向语言表征。基于BERT预训练模型在下游任务中fine-tune可以得到极其优秀的表现，而BERT同样提供了针对一个句子对的编码方法，所以这个文本匹配任务可以容易地通过BERT+Linear的方式解决。

而RoBERTa[2]则是在BERT基础上做了大量优化和调整，比如增加训练时间、增大batch、移除了NSP loss等等，使其相比于BERT具有更强的性能。

由于完成了交互式模型后，我的代码性能技不如人，所以我开始查找近几年文本匹配的paper，我发现最近的一些工作都致力于优化BERT representation，使得同义的文本具有高相似度，不同义的则相似度尽量低。其中的代表便是Sentence-BERT和我选择实现的CoSENT。

CoSENT[3]是苏剑林在今年初在博客发表的文章CoSENT：比Sentence-BERT更有效的句向量方案中提出的一种特征式匹配模型。作者分析了Sentence-BERT保守诟病的问题：训练和预测不一致、调优困难。通过学习SimCSE的针对三元组优化cos similarity的思路，他提出了如下损失函数的计算方法：

如果我们能正确区分句子对，那么所有正样本对的cos肯定需要大于负样本对的cos，这样才能被分开。也就是任意的正样本对(u_i, u_j)和负样本对(u_k, u_l)，都有

$$\cos(u_i, u_j) > \cos(u_k, u_l)$$

考虑用Circle Loss解决这一需求：

$$\mathcal{L} = \log\left(1 + \sum e^{\lambda(\cos(u_k,u_l)-\cos(u_i,u_j))}\right)$$

其中$\lambda > 0$是一个超参数，后面的实验和作者一样取了20。CoSENT的核心就是这个优化cos值的损失函数。

（不过后来我才发现特征匹配模型表现是不如交互式的QAQ...它可能在文本检索的时候更有用一点）

My Implementation

这一节我就来讲讲我的模型性能是如何优化的吧。最开始我直接过一个BERT对两个语句分别求出768维的BERT representation，并直接将两者合并过一个linear层，得到最终结果，这一结果非常差劲，最终结果低于70%，现在想想这样也可以算是一种带弱交互的双塔模型。

之后我改用完全交互式的方法，使用BERT同时编码一个句子对，同样是直接过一个linear层，修改后结果提升到了73.451%。

然后我在此基础上进一步优化，首先增加了Data Preprocessing中数据增强的手段，同时在linear层之前增加了一个dropout层，进行这些改动后提升较为显著，表现达到了80.088%。

之后使用更强大的RoBERTa替换BERT，并选用更小的batch_size增加随机性，使用不断调整的learning_rate等手段，将表现提升到了86.725%。

然后考虑除了pool的结果以外增加hidden state的均值也作为linear的输入，最终提升到了88.053%。

另外我也尝试了XLNet，不过由于其训练时间太长，没有时间进一步优化，导致结果一般(77.876%)就不详说了。

而对于特征式匹配模型，我实现了表现比Sentence-BERT更优秀的CoSENT，不同于交互式模型，特征式模型是双塔结构，即句子对会返回两个各自的向量。根据CoSENT设计的loss，我们可以有效调整BERT representation使得不同义的句子对之间的cos similarity尽量低。最终测试时我们将搜寻一个最佳的threshold将句子间的cos similarity变为答案yes或no。我的实现配合上一步调好的RoBERTa，最终可以得到76.991%的结果。

实验代码：https://github.com/wtl666wtl/Text-Matching

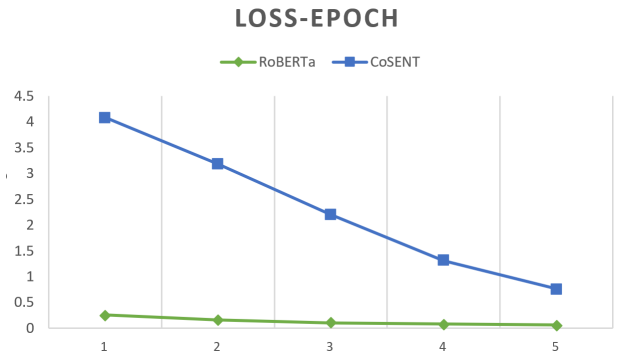
Performance & Analysis

各种实现的最终表现如下表所示：

Method	BERT (w/o data augmentation)	BERT	RoBERTa	CoSENT
Performance	73.451%	80.088%	88.053%	76.991%

可以看出RoBERTa相比于BERT确实优势比较明显。同时可以发现特征式方法明显还是差于交互式了，CoSENT中我完全沿用了RoBERTa的设置，但结果相差接近10%。分析整个训练过程，尤其是loss信息(下图记录了两者在5个epoch内的loss变化)，我认为其中一个原因是CoSENT需要更长的训练周期才达到相同的水平，由于交互式模型可以早早地融合句子对之间的信息，它可以在极少的epoch内将loss压倒很低，而特征式学习仅在最后一步进行融合，想要优化整个representation的计算需要更多的时间(loss)。同时最后我找到的threshold相对较大为0.67，可能也意味着训练还没有完全收敛。作者也对于这个方法有一些思考，他认为对于字面重叠度非常高但语义却不同的对抗性样本，此法会有些吃力，他写道：

> 神经网络本身就是一个连续函数，然后编码器负责将句子压缩到一个句向量中，其结果的连续性必然是非常好的，这里的连续性，指的是句子的微小改动，导致句向量的改动也是微小的；同时，cos的连续性也非常好，即如果Δv比较小，那么$\cos(u, v)$和$\cos(u, v + \Delta v)$的差距也很小。所以，总的来说就是“特征式”的方案连续性会非常好。但问题是，人设计出来的语言天然存在对抗性，即字面上的微小改动能导致标注结果的巨大变化，经典的就是加个“不”字导致所谓的“语义反转”，说白了就是连续性并不好。
>
> 于是，在此类任务之下，连续性非常好的“特征式”方案要去拟合对抗性明显的数据集，就会非常困难。



因此虽然CoSENT理论上可以拟合，但需要花费更多的epoch进行训练，同时有较高的过拟合风险，正是这些原因导致其表现不如交互式方法。

在优化方面，可以看到通过交换次序这种数据增强带来的受益也非常大(提升>6%)，因为这样产生的样本质量很高。我同样测试了在此基础上，增加随机采样得到负样本和通过重复得到正样本的增强方式，但仅提升到了80.973%(提升<1%)，相比于3倍的计算时间显得收效甚微。此外调整batch_size和learning_rate也明显发挥了积极作用，防止整个训练过早地收敛。

另外一个有趣的事情是，我最好的结果只训练了3个epoch，但却比5个epoch的情况要好，但这好像并不普遍，我尝试了一下别的情况下一般5个还是比3个要好的。

Conclusion

在这个文本匹配大作业中，我尝试使用交互式和特征式模型解决这个问题，并通过增加多种优化提升了他们的表现。通过分析，发现交互式模型在此类匹配任务中仍然是较有优势的。我觉得这个问题一个有趣的研究方向是能否把这两个方法在一定程度上的进行合并，各取所长，以达到更好的表现。

Reference

[1] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[2] Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized bert pretraining approach[J]. arXiv preprint arXiv:1907.11692, 2019.

[3] CoSENT: https://www.kexue.fm/archives/8847