自然语言处理大作业报告

苟琪
Nanjing University AI School
MG21370009

摘要

本次 NLP 课程大作业是 Data Fountain 上的一道竞赛题目（评论观点提取），其主要任务有两个。第一是情感分析，其主要目的是预测所给评论文本的情感分类，其中情感分为 3 类（1: 正面，0：负面，2：中立），评价指标是 kappa。第二是命名实体识别，其主要目的是找出所给评论文本中的各种命名实体，实体包括银行实体，产品实体，用户评论实体（形容词），用户评论实体（名词），其标注方式采用 BIO 标注，即 Begin, In, Out 格式。评价指标是 F1 值。

对于第一个任务，其重点应该是样本类别不均衡时的处理。在不增加数据的情况下，我尝试了三种办法，过采样，带权重的交叉熵，和使用 Focal Loss, 但令我颇为遗憾的是三种方式都对我的最终结果没有任何提升反而使其准确率降低，具体实现和分析会在下文中详细介绍。对于第二个任务，NER 是一个较为成熟的 NLP 任务了，目前的主流方法是在分类器后面接 CRF（条件随机场）层，故这部分我也是采用了这部分思想，这个任务我的重点是尝试了几种模型，如 LSTM+CRF,BERT+CRF 和只用 BERT + CrossEntropy 做 NER 任务。虽然可以推测 BERT+CRF 效果最好，但我还是将三种方法的精度进行了对比。

1. 作业内容

本次 NLP 课程大作业是 Data Fountain 上的一道竞赛题目（评论观点提取），其主要任务有两个。

a. 情感分析，其主要目的是预测所给评论文本的情感分类，其中情感分为 3 类（1: 正面，0：负面，2：中立），评价指标是 kappa。

b. 命名实体识别，其主要目的是找出所给评论文本中的各种命名实体，实体包括银行实体，产品实体，用户评论实体（形容词），用户评论实体（名词），其标注方式采用 BIO 标注，即 Begin, In, Out 格式。评价指标是 F1 值。

2. 问题分析

2.1. 情感分类任务

对于情感分类任务，可以选择分类模型有很多，如传统的统计机器学习方法或者 LSTM,CNN，预训练模型等。我选择用 bert + fine-tuning 的方式去做情感分类，但是查看训练数据，发现其类别标注很不平衡，大概比例是 2:0:1 (中立，负面，正面)= 5156:636:230 = 22.4：2.7：1，这会导致模型偏向于预测中立从而影响整体实验结果。这也是实验结果用 kappa 值作为标准的原因，因为如果单纯看正确率的话，模型全预测为中立也会有不错的得分，但是这样没有意义。所以需要对这种不类别不平衡的问题进行处理。
在不增加新的数据的情况下，大致有如下几种处理方式：

- 采样 (过采样和欠采样)
  过采样是指对于类别平衡少的样本进行重复采样，以缓解类别不平衡的问题，欠采样是指从类

别较多的样本中采样出一部分样本，从而抛弃一部分样本来缓解类别不平衡。简而言之，过采样就是重复类别少的样本，欠采样就是抛弃一部分类别比例大的样本，两者都会改变样本的分布，前者会使得模型过拟合类别比例低的样本，后者会使得模型欠拟合类别比例高的样本。

- 修改损失函数
  该问题是个三分类问题，最常用的是交叉熵损失函数，如果要改进的话，可以在计算损失函数的时候赋予每一类样本的权重，这个权重最直观的应该是类别比例小的权重高，类别比例大的样本权重小，从而使模型更多的注意到类别比例低的样本。
  也可以参照 Focal Loss, 这个 loss 也是解决类别不平衡问题所提出的，如下图所示，其有两个参数，γ 和 α, 对于其一般的解释是，α 用于调节样本的类别分布，γ 用于调节简单样本权重降低的速率，就是使模型更多得关注那些难训练的样本, 可以看到当 p_t 接近于 1 的时候，$(1-p_t)^\gamma$ 接近于 0，然后让这部分损失趋近于 0，反过来则损失会被放大，从而达到均衡类别的效果。

$$Fl(pt) = -\alpha * (1-p_t)^\gamma * \log p_t$$

2.2. NER 任务

也有很多选择，可以用 LSTM,BERT 直接当作分类任务处理，当然为了效果更好，我们一般会在后面接 CRF 层，CRF 很适合处理这种任务，可以避免出现以 I-开头的标签出现。在这个任务中，我尝试了很多种模型，如 LSTM+CRF,BERT,BERT+CRF, 然后调参数使模型达到最好的效果。

3. 具体实现

由于本次作业是对中文文本进行处理,这里预训练模型我选择用百度的 Ernie 模型，要让 Epytorch 使用 Ernie, 还需要进行一个模型参数的转化。

3.1. 情感分类

情感分类：Ernie + 分类器（MLP）+ CrossEntropy。然后后续就是一些尝试。

1. 尝试过采样，这里我觉得欠采样会丢失样本，导致本来就不多的样本更少了，所以我决定重复采样一些类别比例低的样本加入训练样本，然后可以选择性的加上正则项。
2. 尝试带权重的交叉熵，这里权重简单的设置为样本数目的倒数。
3. 尝试使用 Focal Loss 来缓解类别不平衡的问题。

3.2. NER

对于 NER 任务，其实这里模型应该是没有任何悬念的，原则上肯定是 Ernie + CRF 效果最好，但是这里我也实现了其它两种，实现了 BiLSTM + CRF, 这里本来也不打算好好训练 LSTM，词向量就随便用的一个，没用最新的，用的是 fast text 词向量。单纯用 bert 也实现了，发现效果还不错，可能是 bert 的编码能力太强了。然后这里的 CRF 我自己也写了一个，相当于我有两个 CRF 版本，一个是官方给的 baseline 改的（官方给的 baseline 不支持 Batch 训练，需要修改下），然后我看了许多资料和博客的公式推导 [1] 总感觉和官方代码有一点出入，然后自己照着实现了一个 CRF 层，最大区别有两点第一是官方给的版本的转移矩阵是从列转移到行，我看着别扭，把它改成了转移矩阵表示从行转移到列的。第二是官方给的版本算 forward score 的时候没有并行计算，我把这部分实现了 batch 计算，少了一个循环，理论上时间复杂度会更低，从实验结果来看也是这样的,因此我最终交的版本是自己写的 CRF 层。

然后中途需要注意的细节还是挺多的，如 bert 分词时原则上对于中文来说应该是按字拆分的，但是我在实现的时候发现有的时候还是出现了长度不一致的情况，最可能的原因是输入文本中有中英混合或者带数字等等情况，这就导致了可能会对英文使用 wordpiece 等分词方式，因此最好的方式是拆分成每个字符再送入 tokenizer 就可以完全保证输入长度和输出长度一致，当然也还需要考虑 [CLS] 和 [SEP] 等特殊符号的处理。

model	score
CrossEntropy	0.7156
过采样	0.6865
Weighted CrossEntropy	0.6934
Focal Loss	0.6786

表 1. 情感分类实验结果

model	score
Ernie + CRF	0.7156
Ernie + CRF2(official)	0.7119
Ernie + CrossEntropy	0.7062
LSTM + CRF	0.6875

表 2. NER 实验结果

4. 实验结果

4.1. 情感分类

情感分类的实验结果如1所示，此处实验结果是固定住 NER 最好的结果然后切换情感分类模型得到的评分。

情感分类实验结果让我很尴尬，因为其一些改进在本问题上完全没有效果甚至都出现了得分降低的情况，原则上采样或者改进 loss 函数会有细微的提升，但是我尝试训练多次效果反而下降。由于时间原因，我可能没有调整到最优的参数，也有可能是我加的方式不对，但是由于要修改的地方不多，我觉得更大可能是这些方法不适用于本次问题。

4.2. NER

NER 效果 Ernie + CRF > Ernie + CRF2 > Ernie + CrossEntropy > LSTM + CRF, 其实验结果如2所示, 此处实验结果是固定住情感分类任务结果后然后送入评分系统所得出得分。从2可以看出来，Ernie + CRF 层效果是最好的，同时单纯用 Ernie 的效果比 Ernie+CRF 略低但也很高，充分说明预训练模型的威力，然后 LSTM + CRF 的效果虽然比前两个低，但是也很不错了。

5. 代码说明及复现

本次作业我将两个任务写在同一个文件下，其中 python sentiment_task.py 是用于训练情感分类任务模型的，python ner_task.py 用于训练 ner 任务，同时在训练好两个模型之后，运行 python predict.py 即可实现预测，在这个文件中，我将两个任务的预测结果整合到一起，并最终输出为以我学号命名的 csv 文件。

对于情感分类任务，我的其它变种的尝试都放在了 sentiment_task.py 中，可以自己选择尝试其它方法进行训练和预测。

对于 ner 任务，我实现的模型都放在了 model.py 中，其训练所包含的类都在 trainer.py 中，如果想要尝试训练不同的模型，可以在 ner_task.py 中改变 model 即可。可以选择的模型有 Bilstm + CRF 和 Ernie + CrossEntropy 和 Ernie + CRF2。其中 CRF 是我自己写的 CRF 版本，CRF2 是改自官方 baseline 所提供的版本，最终提交的版本是用的自己的 CRF 层，训练超参数都是我目前最优的参数。

6. 总结

本次作业到目前为止，我的最好分数是 0.7156，和榜首的分数还有很大差距，我感觉可能有如下原因：参数没调到最优，他们使用了更强的预训练模型，他对不平衡样本用了一些其它 trick，我尝试了三种改进都没有效果甚至分数降低，也有可能是我参数没调好。其实我还是跑了很多次模型和改进，但是没有效果。加上期末时间紧凑，报告写得比较粗糙，latex 排版也不太熟，如以后有时间再改进或学习一下其他人的方法。

参考文献

[1] https://zhuanlan.zhihu.com/p/119254570, 2020. [Online;]. 410