研究生《深度学习》课程

实验报告

实验名称：	综合实验(大作业)
姓　　名：	郑楚彬
学　　号：	21140129
上课类型：	专业课
日　　期：	2021.09.18

一、实验内容

题目二:疫情微博情绪分类

二、实验设计

整体流程如下：

1. 预处理

过滤停用词：采用正则表达式删除如“//@xx:”此类干扰原始文本的词语

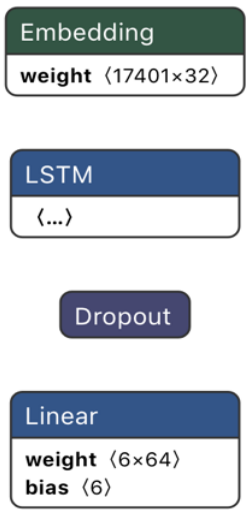
分词：使用 JieBa 分词

训练词/字向量：使用 gensim

构建嵌入层：使用训练好的词/句向量构建嵌入矩阵

构建训练语料：根据嵌入矩阵矩，对文章内容、标签进行编码

2. 模型构建与训练



3. 模型评估

本任务以**宏精准率(macro_P)**、**宏召回率(macro_R)** 、**宏F1值(macro_F1)**作为评测指标

$$macro\_P = \frac{1}{n}\sum_{i=1}^{n} P_i$$

$$macro\_R = \frac{1}{n}\sum_{i=1}^{n} R_i$$

$$macro\_F1 = \frac{2\times macro\_P\times macro\_R}{macro\_P - macro\_R}$$

其中，$n=6$，为情绪类别数 $P_i = \frac{TP_i}{TP_i + FP_i}$ $R_i = \frac{TP_i}{TP_i + FN_i}$

三、实验环境及实验数据集

实验环境：

macOs10.13.6、Pytorch 1.6.0、Jupyter Notebook

数据集：**疫情微博情绪分类**

□ 数据集标签，每条微博被标注为以下六个类别之一:

neural(无情绪)、happy(积极)、angry(愤怒)、

sad(悲伤)、fear(恐惧)、surprise(惊奇)

□ 数据集规模

疫情微博训练数据集包括 6,606 条微博，测试数据集包含 5,000 条微博。

□ 数据集形式

数据集为□格式，包含三个字段:数据编号，文本，情绪标签。

示例: {"id": 11, "content": "武汉加油!中国加油!安徽加油!", "label": "happy"}

□ 数据集预览

```
# data_dir = Path('/Users/zhengchubin/PycharmProjects/learn/data/疫情微博情感分类数据集/')
data_dir = Path('/root/zhengchubin/data/疫情微博情感分类数据集/')
train_file = data_dir.joinpath('virus_train.txt')
test_file = data_dir.joinpath('virus_eval_labeled.txt')

virus_train = json.load(train_file.open(mode='r'))
virus_train = pd.DataFrame.from_records(virus_train)

virus_eval_labeled = json.load(test_file.open(mode='r'))
virus_eval_labeled = pd.DataFrame.from_records(virus_eval_labeled)

print(virus_train.shape, virus_train.shape)
virus_train.head()
```

```
(8606, 3) (8606, 3)
```

	id	content	label
0	1	天使	happy
1	2	致敬[心][心]小凡也要做好防护措施哦//@Mr_凡先生:致敬[心]大家出门记得戴口罩	happy
2	3	[中国赞][中国赞][中国赞]	happy
3	4	悲壮	sad
4	5	！！！一定会好起来	happy

□ 标签对应的文档数量

数据集	积极	无情绪	愤怒	悲伤	恐惧	惊奇	总数
训练集	4423	1460	1322	649	555	197	8606
测试集	923	476	314	165	75	47	2000

四、实验过程

1.1 预处理

□ 过滤停用词/分词

(//@.*?:|@.*?\s|http://[0-9a-zA-Z._?&/]+|[a-zA-Z]+\.[0-9a-zA-Z._?&/]+|#.*?#|
|[\s\n？“、！?!，…”【～】。(_)\[\]\->])

通过如上正则表达式对原始文本进行过滤、分词，得到相对干净的文本序列。并且，以词或字为单位统计个数。

	id	content	label	tokens	tokens_len	chars	chars_len
0	1	天使	happy	[天使]	1	[天, 使]	2
1	2	致敬[心][心]小凡也要做好防护措施哦//@Mr_凡先生:致敬[心]大家出门记得戴口罩	happy	[致敬, 心心, 小凡, 也, 要, 做好, 防护, 措施, 哦, 致敬, 心, 大家, 出...	16	[致, 敬, 心, 心, 小, 凡, 也, 要, 做, 好, 防, 护, 措, 施, 哦, ...	27
2	3	[中国赞][中国赞][中国赞]	happy	[中国, 赞, 中国, 赞, 中国, 赞]	6	[中, 国, 赞, 中, 国, 赞, 中, 国, 赞]	9
3	4	悲壮	sad	[悲壮]	1	[悲, 壮]	2
4	5	！！！一定会好起来	happy	[一定, 会, 好, 起来]	4	[一, 定, 会, 好, 起, 来]	6

可以发现，75%的文章都是短文本（词数<100）。因此，为保证 75%以上的文章在后续的模型训练中能够完全利用到足够的文本新，设置最长词序列长度为 25，最长字序列长度为 45。

	id	tokens_len	chars_len
count	8606.000000	8606.000000	8606.000000
mean	4303.500000	19.954567	35.156286
std	2484.482542	35.575516	63.224908
min	1.000000	0.000000	0.000000
25%	2152.250000	5.000000	8.000000
50%	4303.500000	11.000000	19.000000
75%	6454.750000	24.000000	42.000000
max	8606.000000	1613.000000	2739.000000

□ 训练词/字向量

通过 genism 工具包和清洗好的数据，便可以开始训练我们自己的词向量/字向量模型，向量维度设置为 32，注意此处不引进测试集的数据。

```
word #0/17400 is 的        word #0/3412 is 的
word #1/17400 is 了        word #1/3412 is 一
word #2/17400 is 都        word #2/3412 is 人
word #3/17400 is 我        word #3/3412 is 了
word #4/17400 is 加油      word #4/3412 is 我
word #5/17400 is 在        word #5/3412 is 不
word #6/17400 is 武汉      word #6/3412 is 是
word #7/17400 is 是        word #7/3412 is 们
word #8/17400 is 我们      word #8/3412 is 有
word #9/17400 is 大家      word #9/3412 is 心
```

词/字向量矩阵形状为：基于词 [17401, 32]、基于字 [3413, 32]

□ **构建嵌入矩阵、训练语料**

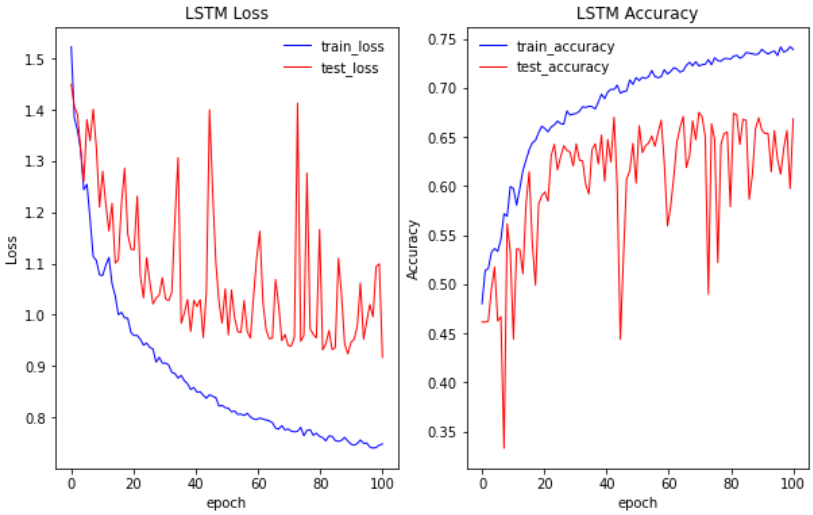
为了方便模型能够对文本进行学习，需要把文本序列按照词向量矩阵相对应的索引进行转化，对于不存在词向量矩阵中的字/词，用**零向量**表示；对于长度不足 25 的需进行补足。示例：

```
tokens:  ['愿', '平安', '致敬', '辛苦', '啦', '愿', '平安']
indices: [30, 14, 22, 41, 526, 30, 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

tokens:  ['马毛', '都', '是', '温暖', '的', '人', '啊']
indices: [10961, 3, 8, 807, 1, 12, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

tokens:  ['正视', '恐惧', '也', '要', '看到', '希望']
indices: [5752, 1500, 17, 11, 156, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

tokens:  ['平安', '平安', '致敬', '伟大', '的', '逆', '行者', '们', '一定', '要', '平平安安', '心']
indices: [14, 14, 22, 119, 1, 694, 690, 53, 40, 11, 116, 18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

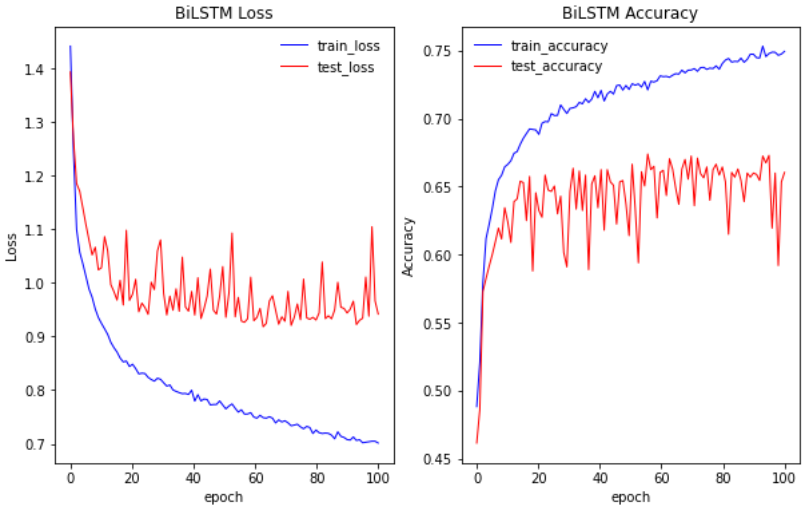
1.2 模型构建与训练

```python
class MyClassificationModel(nn.Module):

    def __init__(self, vector_matrix, hidden_size, output_size,
                 drop_out_rate=0.3, bidirectional=False):
        super(MyClassificationModel, self).__init__()
        self.hidden_size = hidden_size
        self.output_size = output_size
        self.num_layers = 1
        self.bidirectional = bidirectional

        self.embedding = nn.Embedding.from_pretrained(vector_matrix)
        self.lstm = nn.LSTM(
            input_size=self.embedding.embedding_dim,
            batch_first=True,
            hidden_size=hidden_size,
            bidirectional=self.bidirectional,
            num_layers=self.num_layers
        )
        if bidirectional:
            hidden_size = 2 * self.num_layers * hidden_size
        else:
            hidden_size = self.num_layers * hidden_size

        self.drop_out = nn.Dropout(p=drop_out_rate)
        self.softmax = nn.Linear(hidden_size, output_size)

    def forward(self, x: torch.Tensor):

        # 转为词向量 [128, 30, 32]
        output = self.embedding(x)

        # LSTM
        if self.bidirectional:
            # outputs: [128, 30, 2 * 32],
            # h_n:     [2, 128, 32]
            outputs,(h_n, c_n) = self.lstm(output, None)  # (h, c)

            output_fw = h_n[-2, :, :]  # 正向最后一次的输出
            output_bw = h_n[-1, :, :]  # 反向最后一次的输出
            output = torch.cat([output_fw, output_bw], dim=-1)  # [128, 30, 64]

        else:
            # [128, 30, 32]
            output, _ = self.lstm(output, None)  # h

            # 选取最后一个时刻的输出 [128, 32]
            output = output[:, -1, :]

        output = self.drop_out(output)
        output = self.softmax(output)
        return output
```

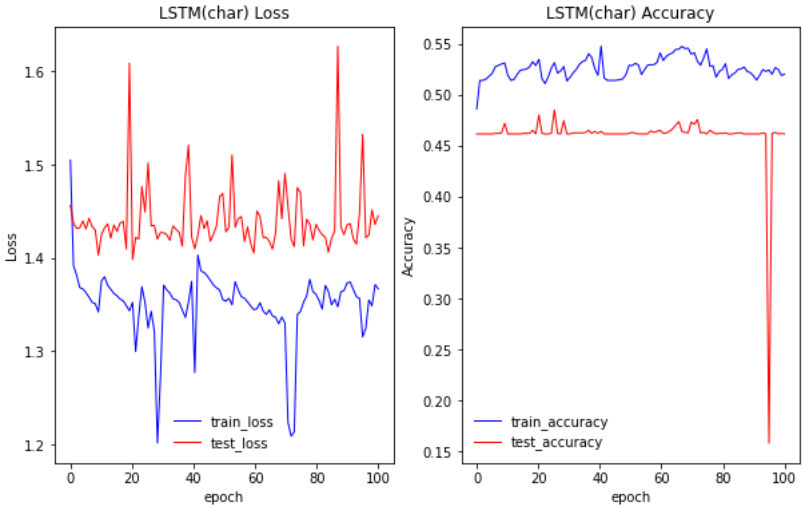
□ 基于词的单向 LSTM 模型



□ 基于词的双向 LSTM 模型



□ 基于字的单向 LSTM 模型



五、实验结果

模型评估 (基于词的双向 LSTM 模型)

□ 精准率、召回率、F1 值

	precision	recall	f1-score	support
happy	0.80	0.81	0.80	923
sad	0.44	0.20	0.28	165
neural	0.62	0.57	0.59	476
fear	0.23	0.33	0.27	75
angry	0.52	0.72	0.60	314
surprise	0.67	0.04	0.08	47

□ 平均准确率：0.653

□ 总结

（1）在短文本场景下，RNN 模型可能不太合适。
（2）存在部分脏数据
（3）不同情感标签的语料不均衡，存在数据倾斜的问题。
（4）在情感分析任务中，基于字的分类模型效果远远不如基于词的。

六、实验心得体会

本次实验加深了对 RNN 相关模型的理解，从零实现了 NLP 领域的情感分析任务，从预处理、分词、生成词向量、模型训练、评估这一整套流程有了一定的认知。虽然整体的模型分类效果不如预期，并且存在很多细节上的问题，但对我而言，收获满满。人工智能的征途是星辰大海，共勉。

七、参考文献

八、附录

需要补充说明的内容，如无可略。

实验报告编写要求

1. 正文要求小四号宋体，行间距 1.5 倍；

2. 英文要求小四号 Times New Roman；

3. 在实验内容、实验过程、实验结果三部分需要针对当次实验不同的实验内容分别填写（模版以实验一为例），实验设计中如有必要也可以分开填写；

4. 实验报告配图的每幅图应有编号和标题，编号和标题应位于图下方处，居中，中文用五号宋体；

5. 表格应为三线表，每个表格应有编号和标题，编号和标题应写在表格上方正中，距正文段前 0.5 倍行距。表格中量与单位之间用“/”分隔，编号与标题中的中文用五号宋体；

6. 图、表、公式、算式等，一律用阿拉伯数字分别依序连续编排序号。其标注形式应便于互相区别，可分别为：图 1、表 2、公式(5)等。