

本周我们主要完成两部分工作，1 对比去年比赛队伍的成绩，精读比赛成绩较好的队伍论文，并对赛题评价指标理解修正 2 在本机配置 tensorflow 和 keras 环境，使用深度学习中的 LSTM 完成非常基础的两级情感分析 positive /negative，其中分析数据来源于 imdb 语料库。

1 该题目是去年的新题目，参赛队伍较少，官网上仅给出四个队伍的论文，其中 GWU 综合成绩最好，cornpittmh 在中文上表现最好。

赛题主要包括三大部分，1holder 和 target 的配对，其中所有的潜在 target 都已经给出，但是之间的 link 需要计算 2 belief 计算 3 sentiment 计算。下面是关于四个队伍做法的一个 overview.

(1) REDESB :

- 关注信念分类
- 贝叶斯方法，各种特征

(2) CUBISM :

- 情感、信念分开两个系统
- 情感：ACA
- 信念：图

(3) Columbia_GWU:

- 提交了全部 24 种情况
- 情感：2 种方法
- 第一种假定源是作者，随机森林，词汇特征等
- 类似于关系抽取
- 信念：假定源是目标，word-tagger

(4) CORNPITTMICH:

- 规则+基于机器学习的方法
- 英：3 个系统，都有 2 个阶段：1)链接预测；2)情感和信念分类
- 中：
 - 情感分类：混合方法，模型(a)LSTM, Weibo 数据,(b)寻找源,(c)规则模型，基于(a)得到最后输出，3 个提交只是(c)的参数不同
 - 信念分类：3 个相同；论坛：规则模型和线性模型；新闻：规则模型

2 其中有队伍使用 LSTM 模型，且深度学习进行情感分析也是现在很常用的方法，出于这种考量，我们决定尝试一下，首先完成基础工作，在本机上搭建 tensorflow 和 keras 环境，学习 keras 官方文档中的 demo 利用 imdb 的数据进行两级情感

分析。

imdb 是英文影评数据库，因此不需要分词，所有的词按照词频高低编码，即 3 编码的即为出现频率第 3 高的词，每一条影评可以按这种方法编码成向量，将这些向量作为原始特征输入，label 即为 1(positive)/0(negative)。利用上述数据在 LSTM 上训练测试。代码如下：

```
#coding:utf-8
#原始数据已经序列成数字，编码表示出现的频率排名，每条评论作为一个序列，统一成
100 的长度，训练

from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from keras.layers import Embedding
from keras.layers import LSTM
from keras.layers import Conv1D, MaxPooling1D, Convolution1D
from keras.datasets import imdb

max_features = 20000#取频率前 20000 的词
maxlen = 100#序列长度最大 100，多的截掉
embedding_size = 128
kernel_size = 5
filters = 64
pool_size = 4
lstm_output_size = 70
batch_size = 30
epochs = 3
#把前 100 个词序列作为特征，不够补 0，多余截断
x_train = sequence.pad_sequences(x_train, maxlen=maxlen)#序列长度大于
maxlen 截断，小则补 0
x_test = sequence.pad_sequences(x_test, maxlen=maxlen)
model = Sequential()
model.add(Embedding(max_features, embedding_size, input_length=maxlen))#
将正整数下标转化为固定大小的向量
```

```

model.add(Dropout(0.25))#随机断开百分比的神经元连接，防止过拟合
model.add(Convolution1D(filters, kernel_size, activation='relu',))
model.add(MaxPooling1D(pool_length=pool_size))
model.add(LSTM(lstm_output_size))
model.add(Dense(1))
model.add(Activation('sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(x_train, y_train, batch_size=batch_size, nb_epoch=epochs,
        validation_data=(x_test, y_test))
score, acc = model.evaluate(x_test, y_test, batch_size=batch_size)
print('Test score:', score)
print('Test accuracy:', acc)

```

模型训练三次，最终结果为

```

24900/25000 [=====>.] - ETA: 0s
24990/25000 [=====>.] - ETA: 0sTest score: 0.39180421633
Test accuracy: 0.845599994612
[Finished in 506.9s]

```

仅进行情感极性分类效果可以，但是本赛题除 sentiment 外，需要判断 belief，获取 hold 和 target 的配对。并且根据上届队伍的做法，情感词汇表等规则占据很大比重，但是现在官方的数据还没有出，我们接下来考虑复现论文中的做法。

3 关于赛题的一些进一步综述

之前在已经对赛题进行了理解，本周阅读了去年 overview 的论文后，对评价指标等方面有进一步的理解。

在评价指标方面，评分基于 4 元组 PST，赋予 partial credit。详细描述如下：

1. 给定预测的 best.xml 和黄金 best.xml，先都转换为 4 元组，具体方法是，对于每个 (source-entity, target-object) 对，在 best.xml 中查找：

- 4 元组中前 3 个未出现，则创建新 4 元组
- 出现，则把 target-mention 插入 provenance-list

2. 首先评价前 3 个域，与黄金 4 元组匹配，按 match-type 顺序等：

(a) 未匹配，false positive

(b) 都匹配，true positive，分数 1

(c)源、目标、态度类型匹配, true positive, 分数 2/3

(d)值和目标匹配, true positive, 分数 2/3

(e)目标和态度类型匹配, true positive, 分数 1/3

(f)未匹配以上 4 条的也为 false positive

3.对于 true positive, 检查 provenance-list, 是否有相同的 target-mention :

(a)全部 provenance 的情况, 召回率-正确率匹配, 得到 f 值, 对原分数伸缩 ;
若是 0 则变 false positive

(b)单个 provenance 的情况 : 有一个匹配则 true positive, 否则 true positive

最终指标 : f-measure, 注意有宏平均, 微平均的使用。