基于 Bert 的文本情感分类

刘健[1+]

[1](南京大学 计算机科学与技术系,江苏省 南京市 210023)

The Text Sentiment Classification Based On Bert

Liu Jian[1+]

[1](Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China)

+ Corresponding author: Phn +86-**-****-****, Fax +86-**-****-****, E-mail:brooksj@foxmail.com, http://www.nju.edu.cn

Received 2019-06-18; Accepted 2019-00-00

Abstract: Sentiment classification is the process of analyzing and reasoning the sentimental subjective text, that is, analyzing the attitude of the speaker and inferring the sentiment category it contains. Traditional machine learning is usually based on traditional algorithms such as SVM, CRF, and information entropy when dealing with sentiment classification problems. Its advantage lies in its ability to model multiple features, but with a single word manually labeled as a feature, and the corpus insufficient is often the bottleneck of performance. The difficulty in sentiment classification of sentences is how to extract the features that are closely related to the sentimental expression in the sentence. The single word that is manually labeled as the feature ignores the contextual semantic information of the word, resulting in the final classification effect is not ideal. In order to solve the problem of feature extraction, it is decided to use Bert(bidirectional encoder representations from transformer), which issued by Google in 2018,to challenge the sentiment multi-classification task in this experiment. The Bert utilizes the powerful feature extraction ability of transformer to learn bidirectional encoder representations of words, which can make better sentiment decisions. The experimental results also show that the effect of using Bert is much better than the baseline using traditional algorithms.

Key words: sentiment classification; Bert; deep learning; word embedding; character embedding

摘 要: 情感分类是对带有感情色彩的主观性文本进行分析、推理的过程,即分析说话人的态度,推断其所包含的情感类别.传统机器学习在处理情感分类问题的时候通常是基于 SVM、CRF、信息熵等传统算法,其优势在于具有对多种特征建模的能力,但要用人工标注的单个词作为特征, 而语料的不足往往就是性能的瓶颈.对句子进行情感分类的难处在于如何抽取到句子中与情感表达紧密相关的特征,以人工标注的单个词作为特征会忽略单词所处的上下文语义信息,导致最终的分类效果不理想.为了解决特征抽取的难题,决定使用 2018 年 Google 提出的文本预训练模型 Bert(bidirectional encoder representations from transformer)来挑战这次实验中的情感多分类任务.Bert 利用 transformer 超强的特征抽取能力来学习词语的双向编码表示,融合了上下文信息的词语编码能更好地进行情感决策.实验结果也表明使用 Bert 的效果要比使用传统算法的 baseline 好很多.

关键词: 情感分类;Bert;深度学习;单词嵌入;字符嵌入

中图法分类号: **** **文献标识码**: A

1 介绍

短信在现代人的通讯中被广泛使用,大量的微表情被用来替代帮助人们更好得表达自己的情感.然而,对于对微表情不熟悉的人来说,找到他想要的那个表情并不是一件简单的事(也许你可以考虑下你的父母以及爷爷奶奶如何使用这些微表情).因此,很有必要设计一个系统来基于短信的内容自动推荐合适的微表情.之前看 Google I/O 大会看到 Google 做的功能类似的一款产品,通过识别短信内容来提供给用户以微表情替代情感相关文本的选项,感觉很有趣.这项任务其实就是 NLP 领域经典的情感分类问题,通过对短信作情感分析来为其匹配一个最合适的微表情类.对这些短信进行情感分类的难点在于: 1.反讽问题,比如"你牛逼你上啊"; 2.领域相关的问题,"我的电脑散热声音很大"、"我家洗衣机声音很大"这些很可能是差评,而"我家音响声音很大"很可能就是好评; 3.网络流行语也会影响情感分析,比如"给力"、"不明觉厉"、"累觉不爱"、"细思极恐"等,这些词利用传统的分词一般都会被切开,而且会影响词性标注,如果想避免只能加入人工干预,修改分词的粒度和词性标注的结果; 4.文本比较短,省略较严重,导致出现歧义或指代错误等,比如"咬死猎人的狗".传统的统计+规则的方法不能很好得解决这些难点,需要引入深度学习强大的特征抽取能力.2018 年 10 月份,Google 公司提出了 NLP 集大成者 Bert 模型[1] .这个模型既引入了 lstm 的双向编码机制同时还采用了 GPT 中的 Transformer 来做特征抽取,具有非常强大的文本特征提取能力,能学习到句子中潜在的句法和语义信息.除此之外,Bert 基于 character-level 做 embedding,就不存在分词以及测试集包含训练集中未出现词的困扰了.这些优点使得 Bert 能够比较好得解决情感分类问题中的一些难点,实验基于 Google 开源的 Bert 预训练好的中文模型做 fine-tuning,最终的实验效果要比采用传统方法得到得 baseline 好很多.

2 相关工作

现有研究已经产生了可用于情感分析多项任务的大量技术,包括监督和无监督方法.在监督方法中,早期论文使用所有监督机器学习方法(如支持向量机、最大熵、朴素贝叶斯等)和特征组合.无监督方法包括使用情感词典、语法分析和句法模式的不同方法.大约十年前,深度学习成为强大的机器学习技术,在很多应用领域产生了当前最优的结果,包括计算机视觉、语音识别、NLP 等.近期将深度学习应用到情感分析也逐渐变得流行.

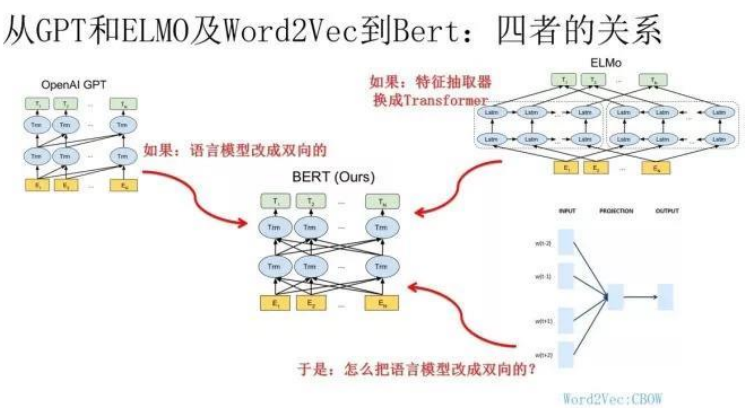

Fig 1 The relationship between Bert and other deep learning models
图 1 Bert 和其它深度学习模型之间的关系

情感分析可划分为 3 种粒度:文档粒度,句子粒度,短语粒度.这次的实验任务主要是基于句子粒度来进行情感分类.Kim 等人于 2013 年提出的 CNN 文本分类工作[2] ,成为句子级情感分类任务的重要 baseline 之一.基本的 lstm 模型加上 pooling 策略构成分类模型,也通常用来做句子级情感分析的方法.Tang 等人于 2015 年发表的工作[3] 使用两种不同的 RNN 网络,结合文本和主题进行情感分析.这几年情感分析方面的突破主要都集中在深度学习领域,深度学习通过学习文本编码表示来抽取文本深层次的特征,解决传统方法无法很好地学习到文本特征的难题.文献[4,8]是从 2013 年至 2018 年期间深度学习在文本特征抽取方面的几项重大成就,其中包

括 Word2Vec,GLoVe,Transformer,ELMo,GPT.而本次实验将采用 Bert 模型[1] 是这几项工作的集大成者,如上图 1 所示是 Bert 与 ELMo 等深度学习模型之间的关系,它结合了 Transformer 和 ELMo 的优点,相比 LSTM 能较好地解决上下文长距离依赖问题,学习到了句子的句法特征以及深层次的语义特征,具有更强的特征抽取能力.由于时间比较紧,就没有拿 Bert 与 ELMo 等其他几个深度学习模型作比较了,但是相比传统的机器学习方法,效果要好很多.

3 方法

3.1 数据预处理

因为 Bert 模型有一个非常重要的超参:输入序列的长度,所以要先确定训练集和测试集中所有句子的最大长度,最终统计得到最长句长为 293,因此将模型最大序列长设为 300 比较合适.如果设得太小模型也不会报错,但是会截断输入从而导致输入信息缺失而不能准确预测所含情感.

除此之外,训练集中的数据并不是均匀分布的,也就是说各个表情所占比重不一样,对各表情聚类统计后发现数据倾斜非常大,最多的表情"心"出现了 74788 次,最少的表情"哈欠"仅出现了 1355 次.为了让 Bert 更好得学习到数据集上的特征,减少数据样本分布不均衡的影响,需要调整不同表情类的损失权重.表情对应样本数越多的,其权重值越小,而表情对应样本数越少的,其权重值越大.具体做法是先统计各表情类的比重,将各表情类的损失值初始化为对应比重,然后将比重最大和比重最小的 2 个表情类的对应损失值置换,再将次大和次小的损失值置换,以此循环直到所有表情类的损失值被置换.

实验任务给出的数据集中不包含验证集,需要从训练集中划分出一个验证集.从训练集每一类中随机划分出等数目的样本,拼凑在一起构成模型的验证集,验证集用于评估模型训练过程中的 F1 score,将 F1 score 最高的几个模型保存到本地,后面预测的时候就可以加载历史保留下来的 F1 score 表现很好的 chechpoint 用于预测.

3.2 Bert模型

使用 Bert 模型的 pytorch 版本库pytorch-pretrained-bert,中文预训练模型使用 Google Search 基于超大规模中文语料库训练的chinese_L-12_H-768_A-12,不过这个预训练模型是提供给 tensorflow 使用的,pytorch 不能直接加载使用,需要先做一个转换.使用 pip 安装好 pytorch-pretrained-bert 之后,在命令行下执行

```
export BERT_BASE_DIR=/path/to/bert/chinese_L-12_H-768_A-12

pytorch_pretrained_bert convert_tf_checkpoint_to_pytorch \
  $BERT_BASE_DIR/bert_model.ckpt \
  $BERT_BASE_DIR/bert_config.json \
  $BERT_BASE_DIR/pytorch_model.bin
```

其中 BERT_BASE_DIR 为下载解压后的预训练模型所在路径.执行完成将在 BERT_BASE_DIR 路径下生成名为 pytorch_model.bin 的 pytorch 可用预训练模型文件,此时对应路径下 bert_model.ckpt 开头的三个文件都可以删除了,因为它们已集成到 pytorch_model.bin 中,对 pytorch 已经没什么用了.

学习率设置也很关键,学习率设置太大不容易收敛到最优值,学习率太小收敛太慢,效率太低.可以在模型开始训练逐渐增大学习率来加快收敛,当增大到一个阈值时就开始减小学习率,减缓梯度变化,让模型更好地落入一个局部最优解.

3.3 baseline模型

baseline 模型使用了传统的机器学习来做,先使用 tf-idf 提取 2000 个特征词,然后每一条短信文本表示成这 2000 个词表示的频率向量,整个训练集和测试集就被转换成维度为样本数*特征词数的频率矩阵.然后使用朴素贝叶斯分类器在频率矩阵上进行训练和预测.

4 实验

4.1 实验环境

```
实验环境:
python 3.6.7
configparser 3.7.4
numpy 1.15.4
tqdm 4.32.1
scikit-learn 0.20.2
torch 1.1.0
torchvision 0.3.0
tensorboradX 1.7
pytorch-pretrained-bert 0.6.2
cuda 9.0.176
cudnn 7.3.0
gpu 型号: 双核 11G 的 Tesla K80(4 块)
```

4.2 实验参数

所有参数设置都写进一个 shell 脚本如下

```
#! /usr/bin/env bash
python sentiment_dev.py \
--data_dir '../input' \
--bert_model_dir '../input/pre_training_models/chinese_L-12_H-768_A-12' \
--output_dir '../output/models/' \
--checkpoint '../output/models/checkpoint-33426' \
--max_seq_length 300 \
--do_predict \
--do_lower_case \
--train_batch_size 60 \
--gradient_accumulation_steps 3 \
--predict_batch_size 15 \
--learning_rate 2e-5 \
--num_train_epochs 3
```

其中 data_dir 为训练集、验证集、测试集所在目录.bert_model_dir 为 Bert 中文预训练模型所在目录.output_dir 为训练模型保存目录以及预测结果输出目录.checkpoint 为模型加载文件,通过指定的 checkpoint 来预测或者从指定 checkpoint 处开始训练以节省训练时间,如果想重新开始训练则去除 checkpoint 这个参数.max_seq_len 为输入序列的最大长度,因为数据集中最长句长为 293,所以这里设为 300 以确保能保留所有输入文本的完整信息.do_predict 表示预测,这个参数改为 do_train 则表示训练.do_lower_case 表示忽略大小写,英文字母都转换成小写.train_batch_size 为一次送入训练的样本数目.gradient_accumulation_steps 为梯度累积次数,每 gradient_accumulation_steps 次梯度计算与反向传播后更改一次学习率,并将梯度归零,实际每次送入训练的样本数目为 train_batch_size/gradient_accumulation_steps 个样本.predict_batch_size 为一次预测的样本数目.learning_rate 为学习率,num_train_epoches 为训练集上的训练轮数.

4.3 实验结果

下图 2 是提交在 kaggle 上的结果截图,其中用红色方框圈出来的结果是提交过的最好结果,这是在训练集上训练 3 轮的结果,test.predict-33426 是在训练集上训练 2 轮半提前终止的结果,submission_2epoch 是在训练集上训练 2 轮的结果.可以看到训练得越久最终的 private score 越高,表明训练程度还不够饱和,还有一些特征没有学习到,需要更深一步的训练.但是训练时耗太长了,没来得及做更多轮的训练作对比.baseline_nb 是用第 3 章提到的 baseline model 得到的结果,与 Bert 得到的结果相差甚远,Bert 预测结果的 F1 score 差不多是 baseline 的 10 倍左右.终榜前提交的最好成绩是 0.18777,榜上排名 12.

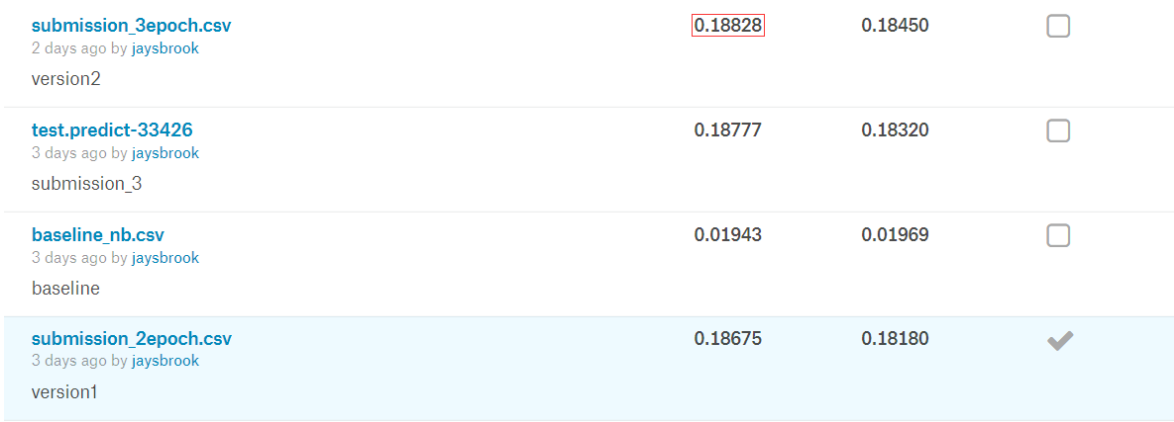

Fig 2 submission results of experiment on kaggle
图 2 实验在 kaggle 上的提交结果

5 总结

文本情感分类是 NLP 领域的一个经典问题,也是 NLP 领域的一个难题.本次实验任务的最大挑战在于它不是简单地判别情感的好与坏,而是需要判别每一条短信中所蕴含的具体情感,然后给它匹配最合适的表情.然而标签数据一共包含 72 类表情,有时候一条短信可能包含不止一种情感,人为判断会觉得可以匹配多类表情,但是数据集中是每一条短信对应一个表情,这无疑给情感分类带来巨大挑战.Bert 可能学习到了短信中的多种情感特征,但是只能给出一种表情,很容易与真实标签不一致.因此 Bert 虽然在这个任务上的表现比 baseline 好很多,但是光看其在测试集上的 F1 score 还是很低的.

本次实验数据预处理的工作做得很少,只做了一些简单的统计,没有对数据进行清洗.训练集中包含大量的非中文文本,而实验中使用的是中文预训练模型,对于非中文文本的编码表示学习效果可能不太好,后续可以引入混合多种语言文本的外部语料库进行训练,提升模型对非中文文本的特征学习能力.

致谢 这学期的数据挖掘课程到此算是结束了.在此,感谢任课老师黎铭老师的辛勤教学,同时感谢这门课的助教在背后的辛勤付出.

References:

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: pre-training of deepbidirectional transformers for language understanding. CoRR, 2018,abs/1810.04805.

[2] Yoon Kim. Convolutional neural networks for sentence classification. CoRR, 2014,abs/1408.5882.

[3] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Target-dependent sentiment classificationwith long short term memory. CoRR,2015,abs/1512.01100.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word represen-tations in vector space. arXiv preprint arXiv, 2013,1301.3781.

[5] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for wordrepresentation. In: Proceedings of the 2014 conference on empirical methods in natural languageprocessing (EMNLP), 2014. 1532–1543.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In: Advances in neural informationprocessing systems, 2017. 5998–6008.

[7] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee,and Luke Zettlemoyer. Deep contextualized word representations. CoRR, 2018,abs/1802.05365.

[8] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving languageunderstanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf, 2018.