1 简介

不管是在机器学习还是深度学习中，目前数据都是核心关键， 好的数据能带领我们得到好的模型，好的模型就能有好的预测结果。 这次我们来看看如果数据不好，会有怎样的结果，如果补充了很多好数据后，模型是不是又可以变好了。

```
from snownlp import SnowNLP

s1 = "我喜欢AI技术"

print("分析得到的正向程度区间在0~1之间，0是最负向，1是最正向。")
print(s1 + " 的正向程度：", SnowNLP(s1).sentiments)

s2 = "我爱AI技术"
print(s2 + " 的正向程度：", SnowNLP(s2).sentiments)

s3 = "我喜欢腾讯技术"
print(s3 + " 的正向程度：", SnowNLP(s3).sentiments)
```

运行结果：

```
分析得到的正向程度区间在0~1之间，0是最负向，1是最正向。
我喜欢AI技术 的正向程度： 0.6127241594259307
我爱AI技术 的正向程度： 0.6689959209445526
我喜欢腾讯技术 的正向程度： 0.49719353330314964
```

2 数据问题

上面我们可以看到这个模型可以区分 "喜欢" 和 "爱" 的差别，认为 "爱" 的正向程度比 "喜欢" 要强烈。 这非常符合我们的预期。但是当我们把 "我喜欢AI技术" 换成 "我喜欢腾讯技术" 的时候，正向分数却降低了很多。 这是为什么呢？

原来一切一切的问题都出在数据集方面。我们知道，模型都有一个训练的过程，只有训练过的模型才有能力预测。

但是如果模型在训练的过程中，没有见过一些特定的数据，但是我们需要预测这样的数据时，模型给出的答案可能就不太准确。 这个解释起来也很容易理解，当你在考试时考的内容是你上课没有学过的内容时，你也答不出来这样的考题。

所以，在上面的例子中，模型的原始训练数据是就要一个书籍购物网站的数据，而这个数据中， 可能并没有 "AI" 或者 "腾讯" 这样的词。所以预测起来也不会很准确。

我给大家展示一下这个模型的原始数据片段。下面这些是消极情感数据片段。

```
如果你在外资公司工作过，尤其你在外资公司担任人力资源的话，就会觉得作者写的东西不是新野，拉拉口中讲的大道理，都是书中有的。作者对人物的描写也太肤浅，真的很失望。
小熊宝宝我觉得孩子不喜欢，能换别的吗
质量还不错，内容只适合边上厕所便消遣了
宝宝一岁七个月,不是太喜欢,也许是因为色彩不够鲜艳.而且对他而言,可能太简单了.
很后悔,不怎么样的一本书,千万别买
没有有价值的技巧和方法。。。网上随便查查信息都比书上的全。
虽然是老乡的作品，但是从客观上来讲的话，还是有点牵强，虽然作者的经历很丰富文采也不错，但不太喜欢结构和风格，这一次跟风跟错了：（
收到的书有断页现象，从 12 页就跳 45 页了
此书我觉得我个人不是特别喜欢，所有的内容太过虚幻了。
```

下面这些是积极情感数据片段。

```
我正在写这本书的心得，勘误和疑点，有兴趣的朋友可以访问我的网站，交流切磋。www.smallstonesoft.com
本人是一名大一学生，大一的生活一直处于浑浑噩噩的状态，直到我看到了这本书。它对于我的意义远远大于一本书。《杜拉拉升职记》让我开始重新审视自己的生活，开始规划自己的未来，在今后，我希望我能向拉拉一样所向无敌。人际关系的交际技巧，与上司下属的巧妙沟通，善于利用学习机会，每一次的金玉良言，我想说，杜拉拉不仅仅给在职的白领以启迪，她势必将改变我的生活。真的谢谢作者。
当时是因为要写读书报告 所以上网到处抄 就发现这本书感觉以前自己没看过中文版的骆驼祥子就看过电影所以决定买个英文的看看本来以为 是中文翻译过来的 不会太地道结果 令我惊讶翻译的非常好！忠实了原著！很多很多复杂的复合句 而且 用词也很得体觉得 是个 非常不错的拓宽知识面的一本小说推荐！
《水知道答案》这本书我一口气买了 5 本，是因为这本书太好了，因为人体的 70%都是水，对着水说爱的语言，水能结出很美丽的结晶，哪么要是人与人之间都说好话，我想我们人类就能很好的控制自己的情绪，人有了一个好心情，干什么都能成功。社会上处处可见“和谐”，人在“和谐”的环境中生活，哪么每一个人不都过上了幸福的生活吗？想自己过幸福生活的人，就去看《小知道答案》吧！
忍不住再说几句，这本书大多内容是参考其他书籍的，例如 P37，内容是参考《金属切削机床》的，参考就参考吧，无可厚非，让我郁闷的试图 3 - 74 的 b)，其中标注 a  b，根本不知道这是干吗的，文中也没有说明。类似的事情，本书太多。我怀疑（是怀疑，没有啥证据）这本书是编辑把握总体框架，教师给素材，研究生做整理。事后没几个人做总体阅读。若没有猜错，第三版也很可能存在这些问题。拭目以待。
阿加莎之外的另一位犯罪推理小说的大师级女作家！女作家和男作家的区别，在于擅长在逻辑严密和气氛紧张的犯罪推理之外，对人性的软弱、复杂和彼此之间可堪或不堪的关系有着细腻的观察和丰富的细节表现。小说更对女性在“法医”这一“男性”职业之种种身心压力、人际压力、规则压力的包围下，如何坚定地维护正义、呵护亲情和爱情，进行了细致入微地表现。
```

3 解决问题

那么怎样让模型预测更加准确呢？答案也很简单，让它学习更多的数据。

snownlp 的训练过程需要先将数据分为正例和负例分别存储。比如正例的句子我们放在 pos.txt 中，每行一条积极的句子，负例放在 neg.txt 中， 每行一条负例句子。

3.1 停用词

在训练之前，还有一点需要注意。句子中不是所有的词都有意义，比如这些标点符号都是可以忽略的，有些非常常见的词也是可以忽略的，像是预期词等。 这些词对于预测的帮助很小，有时候还会因为在训练时出现了太多这样的词而无法注意到那些不常出现，但是一出现就很有代表性的词。

```
--?<>!,."/~-=
只是
只有
只要
只限
叫
叫做
召开
叮咚
可
可以
```

这些常见但不重要的词我们可以在训练时丢弃掉，把它们都放在一个叫 "停用词" 的地方。

下面我们就先加载这些停用词。然后再预测

```
from snownlp import SnowNLP
from snownlp import normal

s3 = "我喜欢腾讯技术"
print(s3 + " 的正向程度：", SnowNLP(s3).sentiments)

print("'只是' 在不在停用词中？", "只是" in normal.stop)

s4 = "只是只是只是我喜欢腾讯技术"
print(s4 + " 的正向程度：", SnowNLP(s4).sentiments)

print("两句正向程度相同？", SnowNLP(s3).sentiments ==
SnowNLP(s4).sentiments)
```

运行结果：

```
我喜欢腾讯技术 的正向程度： 0.49719353330314964
'只是' 在不在停用词中？ True
只是只是只是我喜欢腾讯技术 的正向程度： 0.49719353330314964
两句正向程度相同？ True
```

我们发现，在停用词列表中存在 "只有"。那么如果我们在预测 "只有只有只有我喜欢腾讯技术" 时，计算机就会忽略掉里面所有的 "只有"。 这时这两句话的得分就会是一模一样。

同样，在训练模型的时候，这个停用词也是要预先加载进来的，不然在训练数据的处理阶段，它不会忽略掉这些停用词。 这样对训练也会产生影响。

比如在 "data/test_pos.txt" 中我们加上 "我喜欢腾讯技术；我爱腾讯技术"，训练这些新的数据。

```
from snownlp import SnowNLP
from snownlp import sentiment

def train(neg_file, pos_file, to):
    sentiment.train(neg_file, pos_file)
    sentiment.save(to)

train("data/test_neg.txt", "data/test_pos.txt",
"./data/test_sentiment.marshal")
```

训练好之后，我们的格外数据模型就存到了 "./data/test_sentiment.marshal.3"(注意它比我们给定的 "./data/test_sentiment.marshal" 这个目录多了一个 ".3"). 之后我们就能把这个数据重新加载进来，然后再预测。

4 加载训练数据

下面的代码涉及到了对 snownlp 非常细节的定制化修改。里面使用到了很多特定结构，不要求你掌握这些， 你可以把这个当成一个现成的功能调用就好。

它做的事情简单说就是加载 snownlp 原始模型，然后再将我们刚刚在我们自己数据上训练的模型合并入它原有的模型。

```
import marshal
from snownlp.utils.frequency import AddOneProb
import gzip
from snownlp import sentiment
from snownlp import SnowNLP
```

```
def load_extra_dict(fname, iszip=True):
    sent = sentiment.classifier.classifier
    if not iszip:
        data = marshal.load(open(fname, 'rb'))
    else:
        try:
            data = marshal.loads(gzip.open(fname, 'rb').read())
        except IOError:
            data = marshal.loads(open(fname, 'rb').read())
    for d in data["d"].items():
        if d[0] not in sent.d:
            sent.d[d[0]] = AddOneProb()
        for word, num in d[1]["d"].items():
            sent.d[d[0]].add(word, num)
    sent.total = sum(map(lambda x: sent.d[x].getsum(), sent.d.keys()))

load_extra_dict("./data/test_sentiment.marshal.3")

s3 = "我喜欢腾讯技术"
print(s3 + " 的正向程度：", SnowNLP(s3).sentiments)
```

运行结果：

```
我喜欢腾讯技术 的正向程度： 0.8520710059171597
```

可以看到我们同样是 "我喜欢腾讯技术" 这句话，从原来的 0.4 的分数跃升到了 0.8，变成了一句相对正向的话语。

由此可见，不同数据的作用是非常巨大的。数据工程师的任务就是好好把控好这些数据的质量，为后期的统计，分析，甚至是机器学习做出十分重要的贡献。