CCF BDCI 2018汽车行业用户观点主题及情感识别

贾永辉

**问题简述**：

给定一部分已知的用户对汽车的相关内容文本数据作为训练集，训练集数据已由人工进行了主题和情感标记，然后分析出目标文本所对应的主题和情感。其中训练集数据中主题被分为10类，包括：动力、价格、内饰、配置、安全性、外观、操控、油耗、空间、舒适性。情感分为3类，分别用数字0、1、-1表示中立、正向、负向。

**总体思路：**

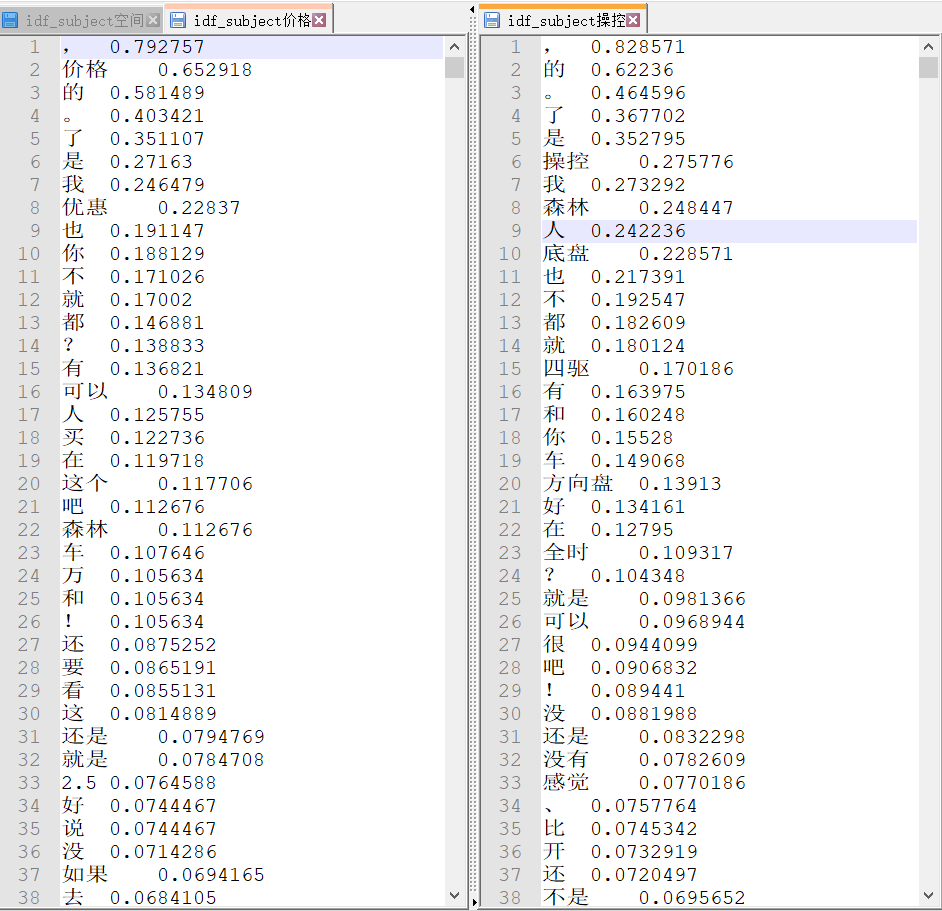
针对汽车行业用户观点主题及情感识别问题，我在处理的时候将其分为了主题分类和情感分类两个独立的问题，然后针对每个问题独立训练提高其各自的准确率进而提高最终结果。

**具体方案与实现细节：**

**主题分类**：

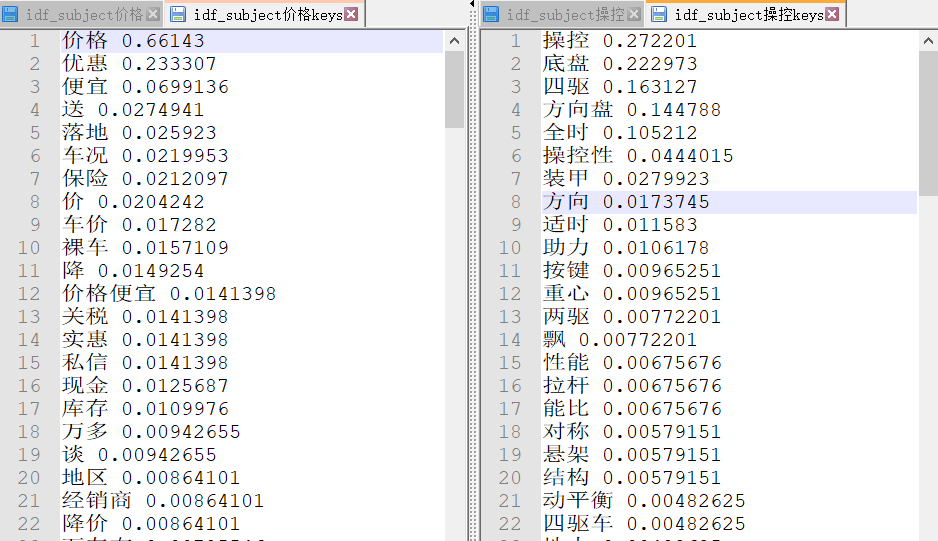
在进行主题分类时，运用的就是libSVM工具，怀着“只要特征选的好，问题就能解得了的”想法，我把主要的心思都放在了数据的处理以及特征的选择上。

在提取特征词的过程中，根据主题的不同，为每种主题都构建了各自的词表，通过计算其idf来显示不同词语在不同主题中出现的频率。部分结果如下：



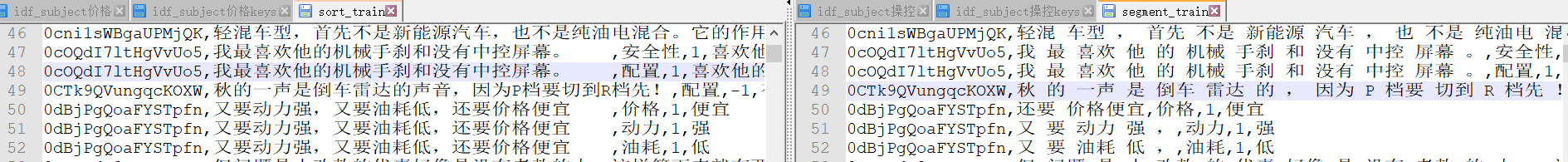
从上面可以看出，一些重要的词，如“价格”、“操控”等都出现在了前面，因为没有去除停用词，所以一些没有明显意义的词也排在了前面。这里提一下为什么我没有首先去除停用词：我最初观察这批数据的时候就发现在这些数据中有些看起来没有价值的词有时候也是有作用的，比如“?”这个标点明显在“中立”这一情感类别中出现的多，因为“?”表示疑问，这种问句基本上都属于“中立”，即0这一类别，所以我尝试不去除停用词，通过构建不同词表来尽可能得到最有价值的特征词。

针对如何确定哪些词是决定类别属性的关键词，我主要想了两种方案：第一种是一个词在哪个类别中出现的最多，则该词属于该类别的特征词，这个得到的词表我也只是用来观察数据，并没有直接用这些词作为特征词；第二种是如果一个词在某一类别中出现的次数远大于在其他类别中出现的次数，则该词是该类别的特征词（实现为：IDF1 \* argument(一个参数) > IDF2+IDF3+IDF4+……），这个得到的结果看起来就算过得去了。



这样初步得到特征词后，我又额外制作了每个主题对应的标准词表，也就是除了满足特征词的词之外，每个主题词表中的其他词都具有相同的idf值（就是全局idf值），然后就额外计算了数据集与每个主题词表的余弦相似度，也将其作为特征进行训练。

运用上面的方法后，我开始着手处理语料中一个句子对应多个类别的情况。针对这种情况，我想出了句子切分的方法：因为一般而言，如果一个句子包含多个主题，那么这个句子中含有的关键字基本上都特别明显，像“动力”，“空间”等。于是我的目标就是将这样的句子一个分为多个，实现的结果如下：



处理之后，一个句子最后便可以得到多个主题，而且句子中的噪音数据也变得更少，然后继续使用前面的基本方法进行训练，结果明显提升。

**情感分类：**

针对情感分类，我再一次运用了上面的基本方法，最后最好的训练时的结果为0.87，但是在提交的结果上并没有比全零好。于是我又查找资料使用了snownlp来进行情感分类，最后的结果仍然一般。根据提交的结果来看，自己实现的情感分类不比直接将情感全置为零的结果好多少。

**To do:**

针对句子切分进行细化，进一步提高切分的准确率和召回率；对于情感分类再选择其他的方式，在这方面还有很大的提升空间。

**附部分成绩提交记录（都是基于前面的思想，可以忽略）：**

2018年10月7日

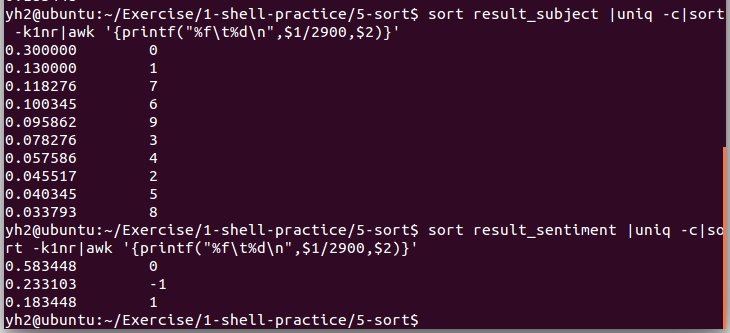
1.目前最好成绩：0.555

方法：情感全零；主题为10个与词典的相似度，10个与keys表的相似度，40个重点词特征。同时对句子进行了主题划分，使一个句子里只剩下一个主题，划分后为2900个句子。

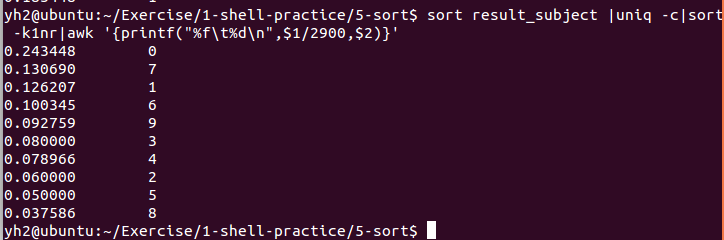
2.与1相比，当划分后的句子为2579时，成绩变成了0.554，说明上面划分多的效果优于现在的。

3.当改用20个词表加727个重点词时，在训练集上准确率达到了0.938462，然而在测试集上相较于1又降为0.5361

-------均衡时，sentiment：3个词表+70个词；subject：不均衡，0个词表+727个词



-------均衡时，subject：0个词表+727个词：



2018年10月8日

为了便于看效果，自己将训练集数据的数据按照3：1又划分为了训练集和测试集；对于主题分类，自己用10个词表和72个词（平衡）作为特征，得到了0.87的准确率；对于情感分析，自己用6个词表加上395个词（不平衡）达到了0.786的准确率，最后的提交结果为0.56

情感分析不变，将主题词表改成每个主题词里面出现大于0.01的词（subject\_word）,270个词(不平衡)，最终结果变为0.563

2018年10月9日

主题分类和情感分类方法不变，只是将test里的数据集降到了2764，最终结果变为0.55929；将test数据集升到3053后，最终结果变为0.5578，所以比较下来还是2900最好，而且即使再更改分句后数据的数目在分句方面的提升也不会很大。

主题分类和情感分类方法不变，将去除关键字函数里的关键词增加后，最终结果变为0.567。

关键词如下：  
temp = [**"动力"**, **"发动机"**, **"性能"**, **"机油"**]  
temp = [**"价格"**, **"便宜"**, **"优惠"**, **"落地"**, **"保险"**]  
temp = [**"内饰"**, **"座椅"**, **"材料"**, **"做工"**, **"细节"**, **"用料"**]  
temp = [**"配置"**, **"高配"**, **"导航"**, **"中控"**, **"雷达"**, **"车载"**]  
temp = [**"安全性"**, **"可靠性"**, **"安全"**, **"气囊"**, **"刹车油"**, **"刹车盘"**]  
temp = [**"外观"**, **"颜值"**, **"车身"**, **"车漆"**, **"前脸"**, **"好看"**, **"丑"**, **"车尾"**, **"难看"**, **"颜色"**, **"漂亮"**, **"色"**]  
temp = [**"操控"**, **"性能"**, **"启停"**, **"四驱"**, **"底盘"**, **"方向"**, **"全时"**]  
temp = [**"油耗"**, **"省油"**, **"费油"**, **"排量"**, **"个油"**]  
temp = [**"空间"**, **"后备箱"**, **"后排"**, **"宽敞"**, **"轴距"**]  
temp = [**"舒适性"**, **"噪音"**, **"空调"**, **"舒适"**, **"风噪"**, **"声音"**, **"响"**]

2018年10月10日

主题分类不变，将情感全设为0，最后结果提升为0.56896，说明情感训练的结果完全没意义，根本比不上全为0的情况。

主题分类方法不变，但是换成了少的类别重复多次的平衡方法而不是多的类别删除的方法，情感仍全为0，最后结果提升为0.57467

201810月11日

主题词表改成只有72词的重点词表后，最后结果降为0.562.