**Archers**

**基于视角的领域情感分析**

**算法说明**

本文对Archers队基于视角的领域情感分析的整体算法做一个说明，算法分为5个部分，分别是数据爬取，汽车领域词典生成，视角抽取，句子预处理，词嵌入训练，情感分析核心算法。下面分别对这5个部分的算法作详细的阐释。

1. **数据爬取**

数据爬取主要是利用网络爬虫技术，采用Python语言中requests，urllib，BeautifulSoup等第三方库，首先利用requests或urllib请求目标网页，得到网页源码的返回，然后对目标网页进行分析，找出目标信息所在的区域，利用BeautifulSoup这款强大的HTML解析工具来获得核心信息，这就是一次爬取的过程。我们可以利用循环，来爬取多个网页的信息，达到大量数据的获取。

数据来源均是公开透明，任何人皆可获取的。

爬取的网站包括：

1.车主之家 （<http://www.16888.com/>）

2.汽车之家 （<http://www.16888.com/>）

3.搜狐汽车 （<http://auto.sohu.com/>）

4.网易汽车 （<http://auto.163.com/>）

1. **汽车领域词典生成**

**2.1处理流程**

从太平洋汽车网，汽车之家以及车主之家网站爬取的初始数据集，经过统计得到一定的规则，利用这些规则来处理生成汽车领域词典。

处理仅供参考，最终结果为final\_user.txt

**2.2类、函数介绍**

Chezhuzhijia\_extract.py, taipingyangcars\_extract.py, qichezhijia\_extract.py分别处理不同的数据

函数的Path表示输入文件的路径。

* chezhuzhijia\_extract.py：处理车主之家数据
* deal\_chezhuzhijia (file\_path,result\_path):

输入：file\_path：输入数据文件名，chezhuzhijiacars.json

输出 ：result\_path: 输出数据文件名

所有的数据均存储在自定义的path路径下。

* taipingyangcars\_extract.py：处理太平洋汽车网的数据
  + deal\_taipingyangcars(file\_path,result\_path)

输入：file\_path：输入数据文件名，taipingyangcars.json

输出 ：result\_path: 输出数据文件名，extract

所有的数据均存储在自定义的path路径下。

* qichezhijia\_extract.py：处理汽车之家数据,得到汽车之家处理后的数据在extract\_qichezhijia.txt中
* 三个网站结果，集合，去重，过滤, 得到最终纯净版视角final\_user.txt

1. **视角抽取**

本部分包含了视角抽取的总体过程，所有过程都基于网络爬虫形成的词典：

**3.1基于多模匹配，利用ViewExtract.java()抽取出两部分**

一部分是单个视角，此类视角由于精确匹配，字面上一定是准确的，该文件输出

为view\_part1.txt

连续的多个视角，此类视角由于连接在一起，极有可能是同一个视角，但也有可能不是同一个视角，因此由基于最长公共子串匹配的python部分程序进一步划分, 该部分文件分为combined\_views.txt

**3.2基于最长公共子串的模式匹配**

利用split\_combined\_views.py ,处理 **1** 当中**b**部分产生的combined\_views.txt，得出view\_part2.txt

将view\_part1.txt和view\_part2.txt组合成最终的generate\_view.txt

将结果进行尾部处理排序输出进入后期的规则处理并形成最终的generate\_view.txt

1. **句子预处理**

**4.1整体说明**

　　predeal\_data.py文件主要实现将训练文本和测试文本转化成具有一定格

　　式，用于后期情感分析任务的处理：

　　训练文本转化的格式为: id\tview\pol\ttext

　　测试文本转化为：id\tview\ttext

　　(id表示记录编号，view表示视角,pol表示情感，text表示相应的句子，\t表示分隔符)

**4.2函数介绍**

* **get\_new\_path (train\_or\_test,isSecond,isSeg):**

根据不同的处理方式设置不同的存储路径，

Train\_or\_test:为0表示是训练集

isSecond:1表示是复赛训练集

isSeg:是否拆分，拆分的意思为对于一个句子中出现多个视角，按语义将文本拆分给不同的视角或者直接把整个句子作为视角的文本，此框架处理采用不需要划分方式。

* **predeal\_data(t\_path,t\_view\_path,t\_data\_result\_path,train\_or\_test,isSecond,isSeg):**

t\_path:表示文本train 或者 test，

t\_view\_path：表示抽取的视角文件

t\_data\_result\_path:结果文件

train\_or\_test:表示处理的是训练集还是测试集，0表示训练集，1表示测试集

isSecond：表示处理的是初赛的数据还是复赛的数据，1表示复赛

isSeg:表示要不要按视角对整个句子进行划分，1表示要划分

* **trans\_space(raw\_path,del\_path,train\_or\_test,isSecond,isSeg):**

采取jieba分词，如果视角中出现空格，要做特殊处理，转化成^号，后面相应的句子中出现的此视角中的空格也为^。

Raw\_path:表示待处理文件

Del\_path:表示结果文件

Train\_or\_path：处理的是训练集还是测试集，0表示训练集，1为测试集

isSecond：表示是否是复赛数据集，0表示初赛数据集

isSeg：表示是否需要切分，0表示不需要，1表示需要

* **filter\_view(text\_path,result\_path,train\_or\_test,isSecond,isSeg)**

根据统计得到的规则，对文本进行视角过滤，过滤不符合规则的视角。

**4.3运行说明**

对于训练集，只需运行：predeal\_data以及trans\_space函数，得到的结果即为训练集结果(采取初赛和复赛训练集结果拼接而成)

* 运行初赛训练数据

predeal\_data('Train.csv', 'Label.csv', 'train\_id\_view\_pol\_t.txt', 0,0,False)  
trans\_space('train\_id\_view\_pol\_t.txt', 'train\_id\_view\_pol\_trans\_t.txt', 0,0,False)

* 运行复赛训练数据

predeal\_data('TrainSecond.csv','LabelSecond.csv','train\_second\_id\_view\_pol\_t.txt', 0,1,False)

trans\_space('train\_second\_id\_view\_pol\_t.txt','train\_second\_id\_view\_pol\_trans\_t.txt', 0,1,False)

* 手动拼接train\_id\_view\_pol\_trans\_t.txt和

train\_second\_id\_view\_pol\_trans\_t.txt文件作为下步算法的输入

* 运行测试集数据

输入的generate\_view.txt为视角抽取结果

predeal\_data('Test.csv','generate\_view.txt','test\_second\_id\_view\_text\_t.txt', 1,0,False)

trans\_space('test\_second\_id\_view\_text\_t.txt','test\_second\_id\_view\_trans\_t.txt', 1,0,False)

filter\_view('test\_second\_id\_view\_trans\_t.txt','test\_second\_id\_view\_t.csv',1,0,False)

得到的‘test\_second\_id\_view\_t.csv’用于下步情感分析处理

1. **词嵌入训练**

在不断改进与优化的过程中，我们做得一大改进就是采用了利用外部数据预训练好的词嵌入模型(词向量模型)，实验证明，采用预训练好的word2vec词向量模型大幅度地提高了算法的精度和泛化表现，效果大大超越了在程序中现场学习词嵌入的效果。

为了不断增加词向量模型的精确度和表现力，我们不仅采用了大规模的互联网上的综合文字数据，如搜狗实验室的互联网全网新闻数据，包含各个类别的新闻，大小超过2G，这有效地训练了非汽车领域词的词向量精确度，为了提高汽车领域的情感分析效果，我们又通过下载和自己爬取等方式，总共收集到了将近500M的汽车语料，总共2.73GB的语料，采用C语言编写的word2vec训练程序进行训练成128维的词向量模型，训练批处理程序train.sh如下：

time ./word2vec -train Train\_segmented.txt -output vectors128.bin -cbow 5 -size 112 -window 8 -negative 25 -hs 0 -sample 1e-4 -threads 20 -binary 1 -iter 15

最后我们直接将vectors128.bin利用到核心算法中去。

1. **情感分析核心算法**

Archers基于视角的领域情感分析核心算法主要分为4个部分，分别是预处理部分，数据变换部分，模型训练与预测部分和结果后处理部分。

**6.1 预处理部分**

本部分将第四部分—句子预处理部分的结果预处理成适合模型输入的数据格式。

**6.1.1输入**

**输入1**： CoreData/train\_id\_view\_pol\_trans\_all.txt

**格式**： id view sentiment sentence (\t分割)

**格式样例**：

*022442 英朗 pos 接下来试驾的英朗，与其他三台而言，豪华的氛围做得最到位，你不能不赞扬通用近来的新款车在材质的选用很有一套，得分！。*

表示训练集ID为022442的句子中视角"英朗"的情绪为pos

**输入2**： CoreData/test\_second\_id\_view\_final.txt

**格式**： id view sentence (\t分割)

**格式样例**：

*100000 雷克萨斯 虽然官方并没有推出什么特别的防弹版本，但雷克萨斯的offroad向来都是私人装甲的热门车型。*

表示测试集ID为100000的橘子，我们识别出了“雷克萨斯”视角，需要对这个句子分析情感，得出这个句子中雷克萨斯的情感极性

**6.1.2输出**

对训练集和测试集进行预处理，得到模型训练所需的特定格式的训练集，以及需要判断情感的测试集。

* 1. **训练集输出**

**输出1**： CoreData/A\_COMMENT200.lbl

**说明**： 训练集的每句话的情感，也是训练集的label文件。

**格式**： sentiment

**格式示例**：

*pos*

第n行为pos表示预处理后的第n个句子的情感为pos

**输出2**： CoreData/A\_LEFT\_COMMENT200.all

**说明**： 句子中位于视角的左边的部分，分好词，\t分开，以便运行左边的GRU。

**格式**：

**格式示例**：

*no 2 柳州 本地 一手 精品*

表示句子“no.2柳州本地一手精品凯越，1.6l手动，4.xx万。”中“凯越”的分好词的左边部分。

**输出3**： CoreData/A\_RIGHT\_COMMENT200.all

**说明**： 句子中位于视角的右边的部分，分好词，\t分开，倒序排列以符合GRU从左到右运行的顺序，以便运行右边的GRU。

**格式**：

**格式示例**：

*万 200 最低*

表示句子“可以买25辆兰博基尼(最低200万)”中“兰博基尼”的倒序的分好词的右边部分。

* 1. **测试集输出**

**输出1**： CoreData/A\_LEFT\_COMMENT200.test

**说明**： 测试集句子中位于视角的左边的部分，分好词，\t分开，以便运行左边的GRU。

**格式**：

**格式示例**：

*no 2 柳州 本地 一手 精品*

表示句子“no.2柳州本地一手精品凯越，1.6l手动，4.xx万。”中“凯越”的分好词的左边部分。

**输出2**： CoreData/A\_RIGHT\_COMMENT200.test

**说明**： 句子中位于视角的右边的部分，分好词，\t分开，倒序排列以符合GRU从左到右运行的顺序，以便运行右边的GRU。

**格式**：

**格式示例**：

*万 200 最低*

表示句子“可以买25辆兰博基尼(最低200万)”中“兰博基尼”的倒序的分好词的右边部分。

注意，如果同一个视角在句子中

**6.2 数据变换部分**

本部分的作用是将前面预处理好的数据，即训练集分好词的左右两部分，以及label，以及测试集的分好词的左右两部分，处理成深度学习模型能够处理的数值，即张量形式。导入前存入的数据张量形式为[None, 100, 1]，None代表每次处理的句子个数是不定的，100表示左边和右边的部分句子长度最大为100，1表示每个句子中的每个词用1维表示。通过数据变换以后，我们得到形式为[None, 100, 128]的张量，即每个词用128维的词向量表示，词向量模型在之前已经预训练完毕。这样的话，数据转换完成，得到了适合深度学习算法处理的数据形式。

**6.3 模型训练与预测**

Archers基于视角的领域情感分析，基本的深度学习模型结构图如图1所示，我们采用一种双向聚焦型GRU(Bi-direction Focus GRU, BFGRU)算法来对特定视角出现在特定位置所代表的情感来分类，GRU单元能够解决纯RNN网络中的梯度消失和梯度爆炸问题，能够对句子信息实现长期记忆，并且是LSTM的简化版本，在本赛题中，数据集并不是很大，初赛和复赛数据集加起来都只有20000多条，不适宜采用较复杂的LSTM模型，也不适宜增加过多的参数，否则很容易产生过拟合，这大概也是双向聚焦型GRU的表现比双向聚焦型LSTM效果要好的原因。

整个算法思想如下，我们找出特定视角V出现的特定位置P，我们将P的左边分好词，替换成词向量，作为左边单向GRU算法的输入，将P的右边部分分好词，替换成词向量，作为右边往左边的单向GRU算法的输入，最终运行到P的左右邻接词时，左右的GRU算法都会得出一个输出，左边的输出代表了左边文字的描述，右边的输出代表了右边文字的描述，我们将两个输出merge到一起，相当于综合了该处视角的上下文信息，对该处视角做一个情感的分析与判断。merge以后利用Softmax进行分类，分为pos,neu,neg三类作为情感预测值。

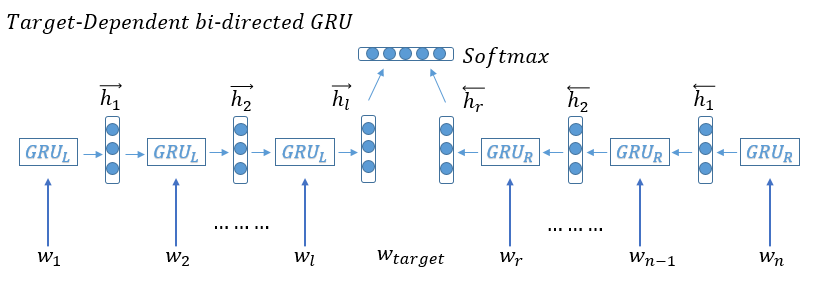


图1. Archers情感分析核心算法示意图

图2是核心算法的一个示例，综合“奔驰”视角的左右描述信息，就得到“奔驰”视角的情感态度，通过大量的语料，算法会学习出“不太 喜欢”这样的描述更可能是neg情感，所以能够对句子中视角的情感做出较为准确的分类。

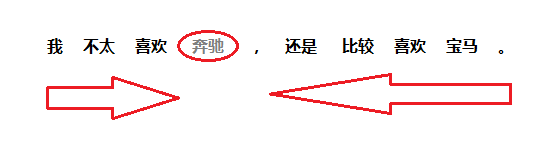
****

图2. Archers情感分析核心算法示例

**6.4 结果后处理部分**

核心算法的预测结果是一个三元组[a, b, c]，分别可以看做属于pos, neu, neg三类情感的概率值，我们需要做一定的后处理，处理成适合比赛要求的提交数据。

在后处理阶段，由于在一个句子中同一个视角出现多次，我们会拆分成不同的左右部分，所以对于一个视角V和一个句子S，假设V在S中出现C次，那么句子S就会被分成C种不同的左右部分，然后相当于视角V和句子S重复出现C次，最后我们需要将V在S中的情感综合考虑，总结出来，这时我们采用的是**加和打分算法**，就是说，每次V,S的重复出现都会得到一个情感分类预测的原始结果，比如[0.2, 0.7, 0.1]，这就代表在句子中某个位置出现的视角V，情感更加倾向于pos，如果在另一个位置情感分类结果为[0.9, 0.05, 0.05]，那么说明在这个位置更加倾向于neu，这时我们综合一下，将各类分数相加，得到[1.1, 0.75, 0.15]，最终得到V在S中的综合情感为neu。这就是我们加和打分算法的思想。

做了这样一个合并之后，我们再将没有出现视角的句子的ID写入到结果文件中，就得到了需要提交的格式的一个结果文件。