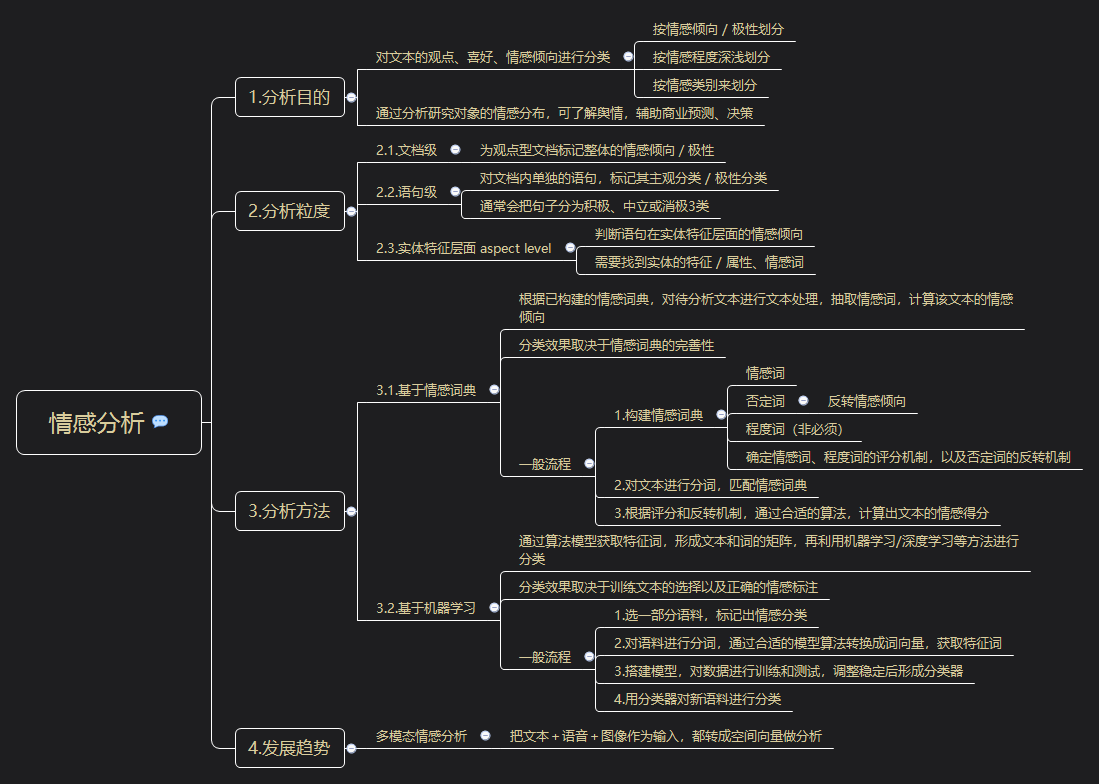
# 1 概述

情感分析是自然语言处理中常见的场景，比如淘宝商品评价，饿了么外卖评价等，对于指导产品更新迭代具有关键性作用。通过情感分析，可以挖掘产品在各个维度的优劣，从而明确如何改进产品。比如对外卖评价，可以分析菜品口味、送达时间、送餐态度、菜品丰富度等多个维度的用户情感指数，从而从各个维度上改进外卖服务。



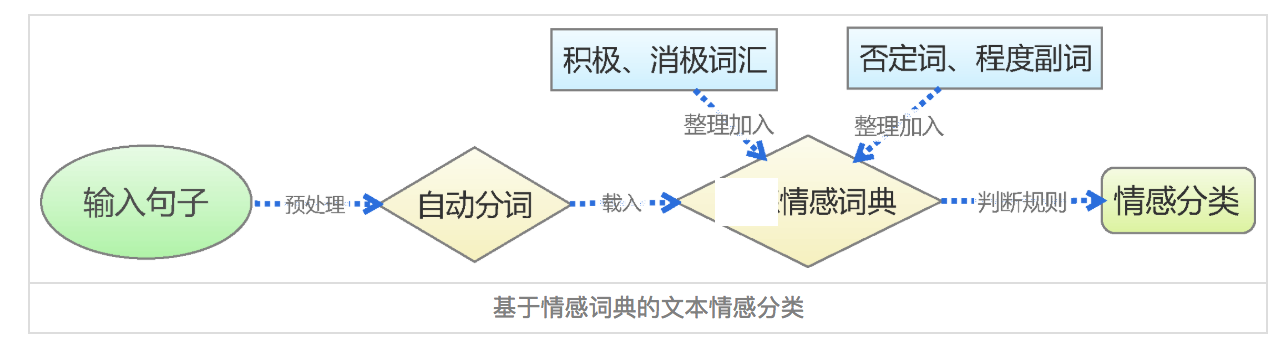
情感分析可以采用基于情感词典的传统方法，也可以采用基于机器学习和深度学习的方法，下面详细讲解

# 2 情感分析方法

## 2.1基于情感词典的传统方法

### (1) 基于词典的情感分类步骤

基于情感词典的方法，先对文本进行分词和停用词处理等预处理，再利用先构建好的情感词典，对文本进行字符串匹配，从而挖掘正面和负面信息。如下图



### (2) 情感词典

情感词典包含正面词语词典、负面词语词典、否定词语词典、程度副词词典等四部分。如下图

词典包含两部分，词语和权重，如下

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 正面： | 负面： | 程度副词： | 否定词： |
| 2 | 很快 1.75 | 无语 2 | 超级 2 | 不 1 |
| 3 | 挺快 1.75 | 醉了 2 | 超 2 | 没 1 |
| 4 | 还好 1.2 | 没法吃 2 | 都 1.75 | 无 1 |
| 5 | 很萌 1.75 | 不好 2 | 还 1.5 | 非 1 |
| 6 | 服务到位 1 | 太差 5 | 实在 1.75 | 莫 1 |
| 7 |  | 太油 2.5 |  | 弗 1 |
| 8 |  | 有些油 1 |  | 毋 1 |
| 9 |  | 咸 1 |  |  |
| 10 |  | 一般 0.5 |  |  |

情感词典在整个情感分析中至关重要，所幸现在有很多开源的情感词典，如BosonNLP情感词典，它是基于微博、新闻、论坛等数据来源构建的情感词典，以及知网情感词典等。当然我们也可以通过语料来自己训练情感词典。

### (3) 分词方法

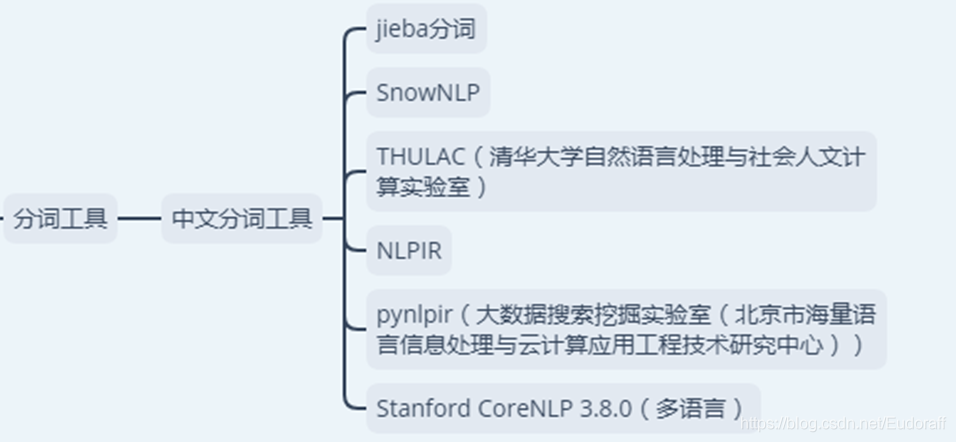
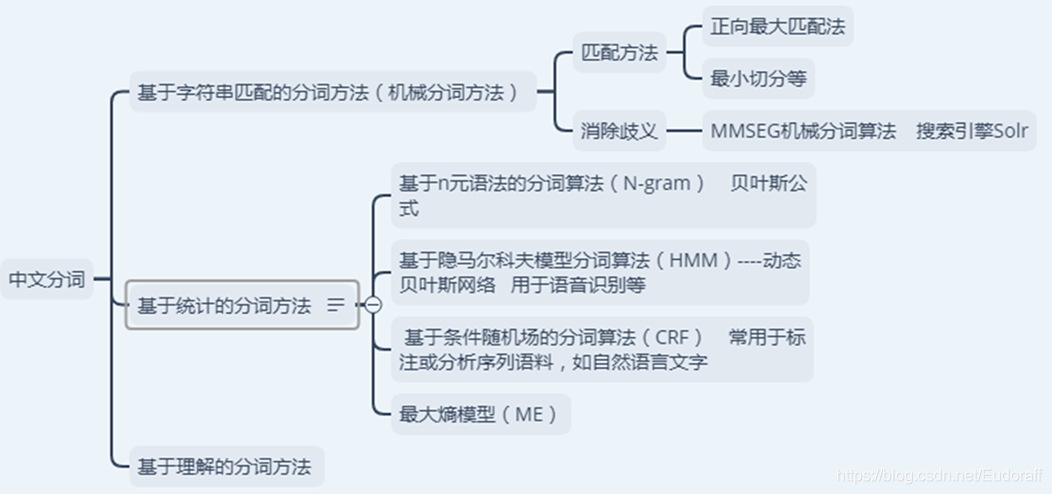
英文分词

（1）根据空格拆分单词（Split）

（2）排除停止词（Stop Word）

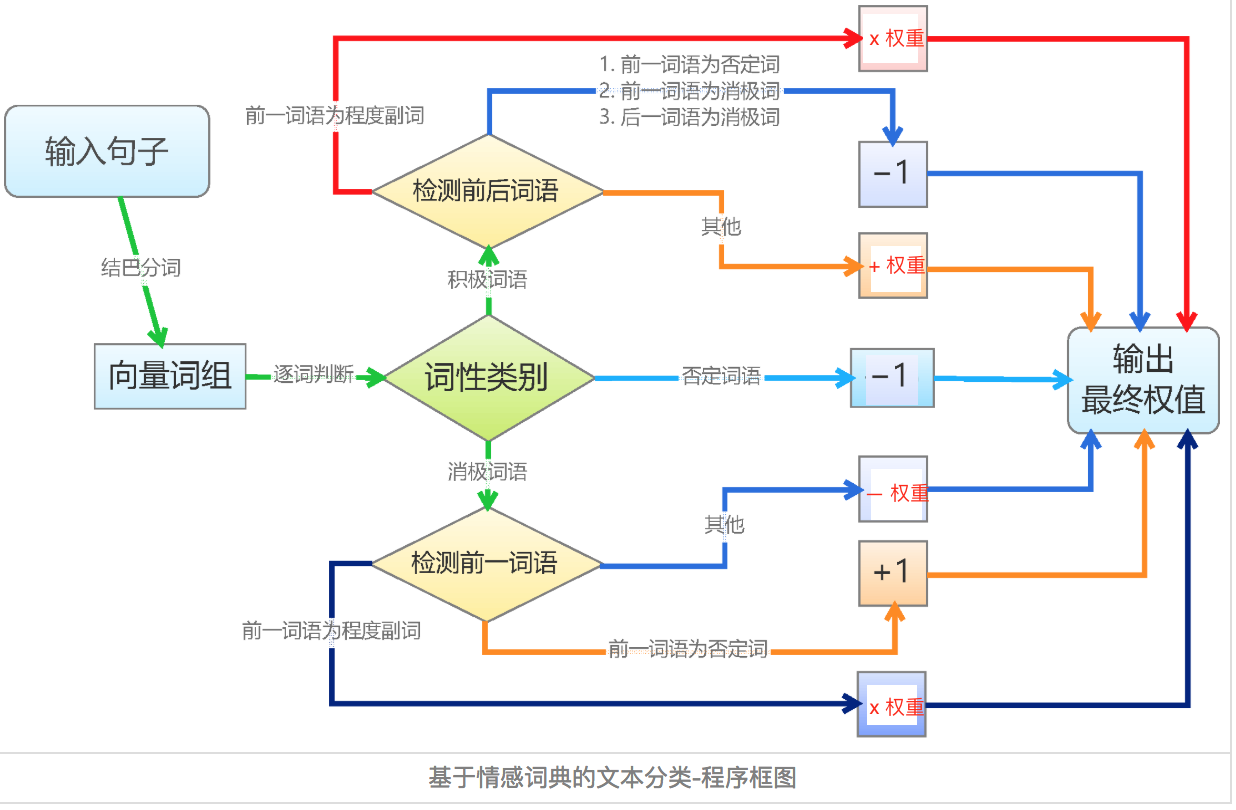
（3）词干提取(stemming)和词形还原(lemmatization)

中文分词



### (4) 情感词典文本匹配算法

基于词典的文本匹配算法相对简单。逐个遍历分词后的语句中的词语，如果词语命中词典，则进行相应权重的处理。正面词权重为加法，负面词权重为减法，否定词权重取相反数，程度副词权重则和它修饰的词语权重相乘。如下图

利用最终输出的权重值，就可以区分是正面、负面还是中性情感了。

### (5) 缺点

基于词典的情感分类，简单易行，而且通用性也能够得到保障。即准确率很高，召回比较低。对于不同的领域，构建情感词典的难度是不一样的，精准构建成本较高。

1、精度不高。语言是一个高度复杂的东西，采用简单的线性叠加显然会造成很大的精度损失。词语权重同样不是一成不变的，而且也难以做到准确。

2、新词发现。对于新的情感词，比如给力，牛逼等等，词典不一定能够覆盖词典构建难。

3、基于词典的情感分类，核心在于情感词典。而情感词典的构建需要有较强的背景知识，需要对语言有较深刻的理解，在分析外语方面会有很大限制。

## 2.2 基于机器学习的算法

基于机器学习的情感分析思路是将情感分析作为一个分类问题来处理，具体的流程如下：

### (1) 文本预处理

特征抽取：中文最小语素是字，但是往往词语才具有更明确的语义信息，但是随着分词，可能出现词语关系丢失的情况。n-元文法正好解决了这个问题，它也是传统机器学习分类任务中最常用的方法。

文本向量化：对抽取出来的特征，向量化是一个很重要的过程，是实现由人可以理解的文本转换为计算机可以处理数据的重要一步。这一步最常用到的就是词袋模型（bag-of-words ）以及最近新出的连续分布词向量模型（word Embedding）。词袋模型长度为整个词表的长度，词语对应维度置为词频，文档的表示往往比较稀疏且维度较高。Embedding的表示方式，能够有效的解决数据稀疏且降维到固定维度，更好的表示语义信息。对于文档表示，词袋模型可以直接叠加，而Embedding的方法可以使用深度学习的方法，通过pooling得到最终表示。

特征选择：在机器学习分类算法的使用过程中，特征好坏直接影响机器的准确率及召回率。选择有利于分类的特征，可以有效的减少训练开支及防止模型过拟合，尤其是数据量较大的情况下，这一部分工作的重要性更加明显。其选择方法为，将所有的训练语料输入，通过一定的方法，选择最有效的特征，主要的方法有卡方，信息熵，dp深层感知器等等。

目前也有一些方法，从比句子粒度更细的层次去识别情感，如基于方面的情感分析（Aspect based Sentiment Analysis），他们从产品的评价属性等更细粒度的方面对评价主体进行情感倾向性分析。

### (2) 分类算法选择

文本转换为机器可处理的结构后，接下来便要选择进行机器学习的分类算法。常用方法：朴素贝叶斯、最大熵、支持向量机、k近邻模型等。

## 2.3 基于深度学习的算法

近年来，深度学习在NLP领域内也是遍地开花。在情感分类领域，我们同样可以采用深度学习方法。基于深度学习的情感分类，具有精度高，通用性强，不需要情感词典等优点。

### (1) 基于深度学习的情感分类步骤

基于深度学习的情感分类，首先对语句进行分词、停用词、简繁转换等预处理，然后进行词向量编码，然后利用LSTM或者GRU等RNN网络进行特征提取，最后通过全连接层和softmax输出每个分类的概率，从而得到情感分类。

### (2) 代码原理

下面通过代码来讲解这个过程。项目数据来源于大众点评，训练数据10万条，验证1万条。分析大众点评用户评论中，关于交通，菜品，服务等20个维度的用户情感指数。分为正面、负面、中性和未提及四类。代码在验证集上，目前f1 socre可以达到0.62。

步骤一： 分词和停用词预处理

数据预处理都放在了PreProcessor类中，主函数是process。步骤如下

1、读取原始csv文件，解析出原始语句和标注

2、错别字，繁简体，拼音，语义不明确等词语的处理

3、stop words停用词处理

4、分词，采用jieba分词进行处理。由于分词后较多口语化的词语不在词向量中，所以对这部分词语从jieba中del掉，然后再进行分词。直到只有为数不多的词语不在词向量中为止。

5、构建词向量到词语的映射，并对词语进行数字编码。这一步比较常规。

步骤二：词向量编码

词向量编码步骤主要有：

1、加载词向量。词向量可以从网上下载或者自己训练。网上下载的词向量获取简单，但往往缺失特定场景的词语。比如大众点评菜品场景下的鱼香肉丝、干锅花菜等词语，而且往往这些词语在特定场景下还十分重要。而自己训练则需要几百G的语料，在高性能服务器上连续训练好几天，成本较高。可以将两种方法结合起来，也就是加载下载好的词向量，然后利用补充语料进行增量训练。

2、建立词语到词向量的映射，也就是找到文本中每个词语的词向量

3、对文本进行词向量编码，可以通过keras的Embedding函数，或者其他深度学习库来搞定。

步骤三：构建LSTM网络

LSTM网络主要分为如下几层

1、两层的LSTM。

2、dropout(随机/临时删掉网络中一半的隐藏神经元，输入输出神经元保持不变，https://blog.csdn.net/program\_developer/article/details/80737724)，防止过拟合

3、全连接，从而可以输出类别

4、softmax，将类别归一化到[0, 1]之间

LSTM网络是重中之重，这儿可以优化的空间很大。比如可以采用更优的双向LSTM，可以加入注意力机制。这两点都可以提高最终准确度。另外可以建立分词和不分词两种情况下的网络，最终合并。

步骤三： softmax输出类别

softmax只是一个归一化，讲数据归一化到[0, 1]之间，从而可以得到每个类别的概率。我们最终取概率最大的即可。

### (3) 基于深度学习的情感分析难点

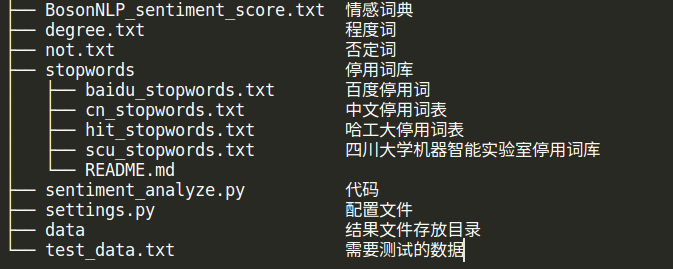
基于深度学习的情感分析难点也很多

1、语句长度太长。很多用户评论都特别长，分词完后也有几百个词语。而对于LSTM，序列过长会导致计算复杂、精度降低等问题。一般解决方法有进行停用词处理，无关词处理等，从而缩减文本长度。或者对文本进行摘要，抽离出语句主要成分。

2、新词和口语化的词语特别多。用户评论语句不像新闻那样规整，新词和口语化的词语特别多。这个问题给分词和词向量带来了很大难度。一般解决方法是分词方面，建立用户词典，从而提高分词准确度。词向量方面，对新词进行增量训练，从而提高新词覆盖率。

# 3 代码实现

## 【目录介绍】



## 3.1 基于情感词典法/根据实际场景切换辞典

1、BosonNLP情感词典

(1) 适用范围：BosonNLP是基于微博、新闻、论坛等数据来源构建的情感词典

(2) 算法过程：

准备文件：BosonNLP情感词典，停用词，程度词，否定词

输入：测试数据

输出：情感分数(>0积极，按实际情况设置value值)

(3) 例：我今天很高兴也非常开心

去除停用词：['今天', '很', '高兴', '非常', '开心']

情感词：{0: '0.239760556816', 2: '1.48950851679', 4: '2.61234173173'}

程度词对应的权重：{1: '2.5', 3: '3'}

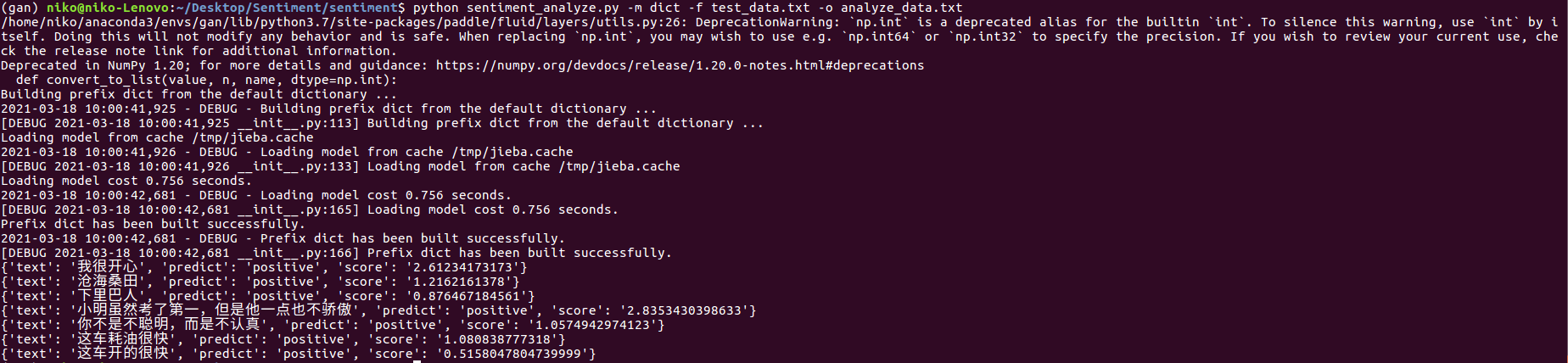
否定词(若有一个否定词就是一个-1)：{}

分数计算：1\*0.239760556816+2.5\*1.48950851679+3\*2.61234173173=23.556094836766

(4) 脚本测试：

执行命令：python sentiment\_analyze.py -m dict -f test\_data.txt -o analyze\_data.txt

输出结果：



(5) 优缺点：

通用性不高，测试速度慢。

## 3.2 Jiagu方法----https://github.com/ownthink/Jiagu

(1) 适用范围：Jiagu使用大规模语料训练而成。将提供中文分词、词性标注、命名实体识别、情感分析、知识图谱关系抽取、关键词抽取、文本摘要、新词发现、情感分析、文本聚类等常用自然语言处理功能。

(2) 例：

import jiagu

text = '我爱中国'

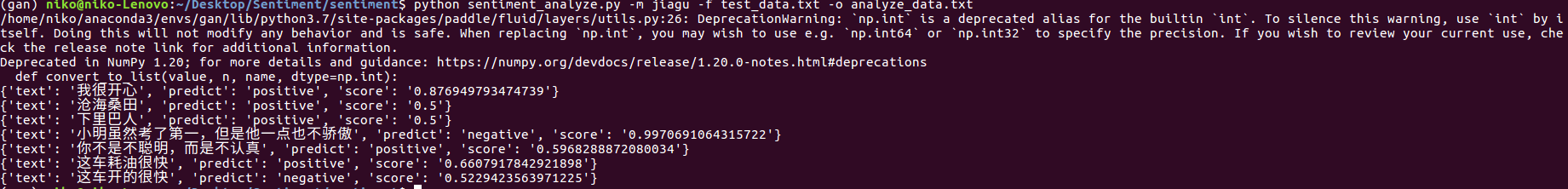
sentiment = jiagu.sentiment(text)

print(sentiment)

(3) 脚本测试：

执行命令：python sentiment\_analyze.py -m jiagu -f test\_data.txt -o analyze\_data.txt

输出结果：



(5) 优缺点：

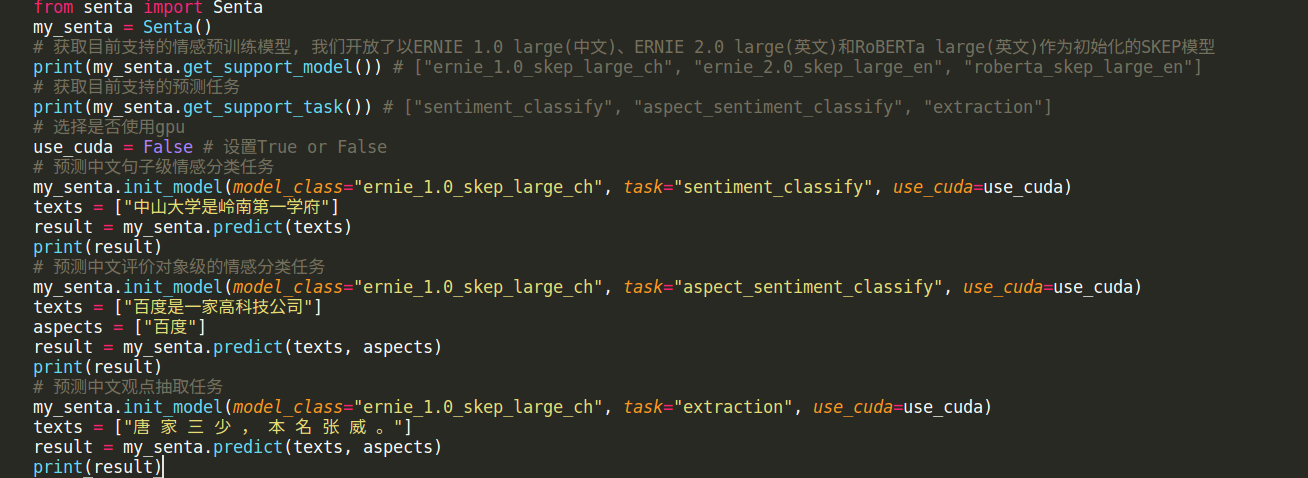
准确率不高，测试速度快。

## 3.3 Senta方法--https://github.com/baidu/Senta

(1) 适用范围：百度在Senta中开源了基于SKEP的情感预训练代码和中英情感预训练模型。而且，为了进一步降低用户的使用门槛，百度在SKEP开源项目中集成了面向产业化的一键式情感分析预测工具。情感分析旨在自动识别和提取文本中的倾向、立场、评价、观点等主观信息。它包含各式各样的任务，比如句子级情感分类、评价对象级情感分类、观点抽取、情绪分类等。

### 3.3.1 一键化工具—安装步骤及使用方法见https://github.com/baidu/Senta

(1) 例：



(2) 脚本测试(运行时间长，对环境要求高)：

执行命令：python sentiment\_analyze.py -m senta -f test\_data.txt -o analyze\_data.txt

### 3.3.2 senta-paddlehub法---利用paddlepaddle(百度研发的开源开放的深度学习平台)--https://mp.weixin.qq.com/s/yyYSBscDcmccWxkRwXRJ8w

(1) 安装步骤：

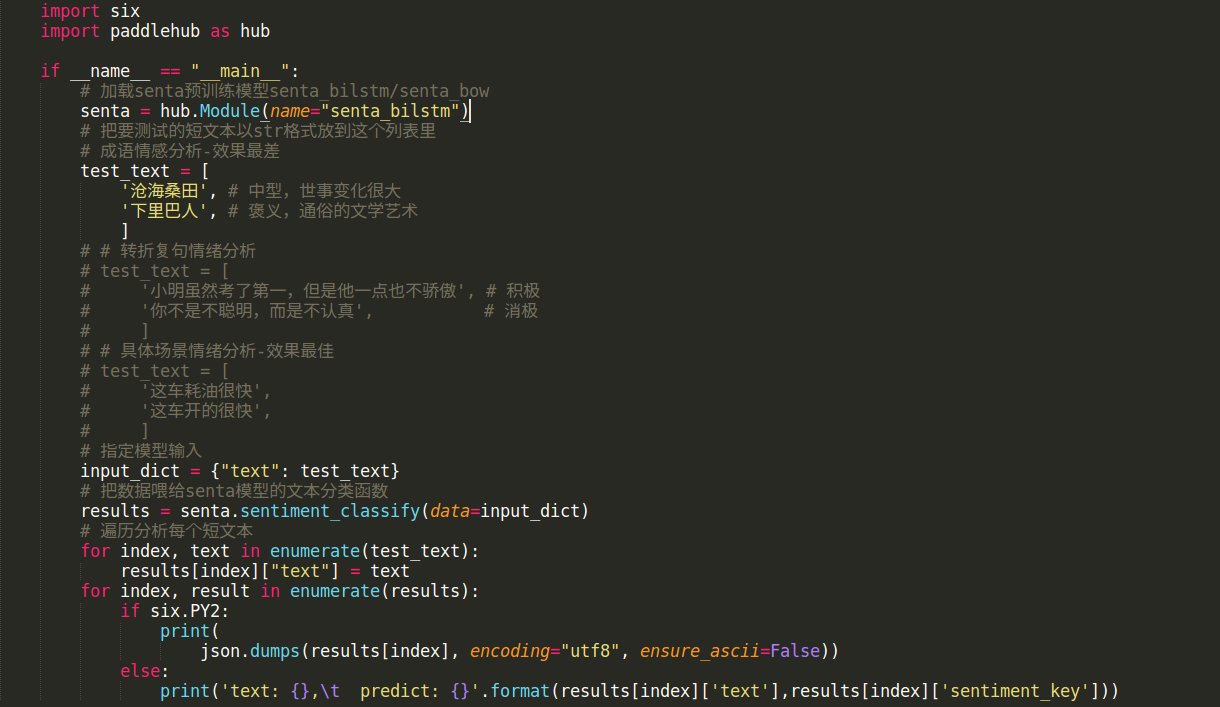
paddlepaddle与tensorflow区别：前者偏于应用直接调用模型，后者偏于研究）

安装步骤：cpu/gpu--https://github.com/PaddlePaddle/PaddleHub/tree/release/v0.5.0/demo/senta（在安装Senta的基础上再安装paddlehub）

pip install paddlepaddle

pip install paddlehub

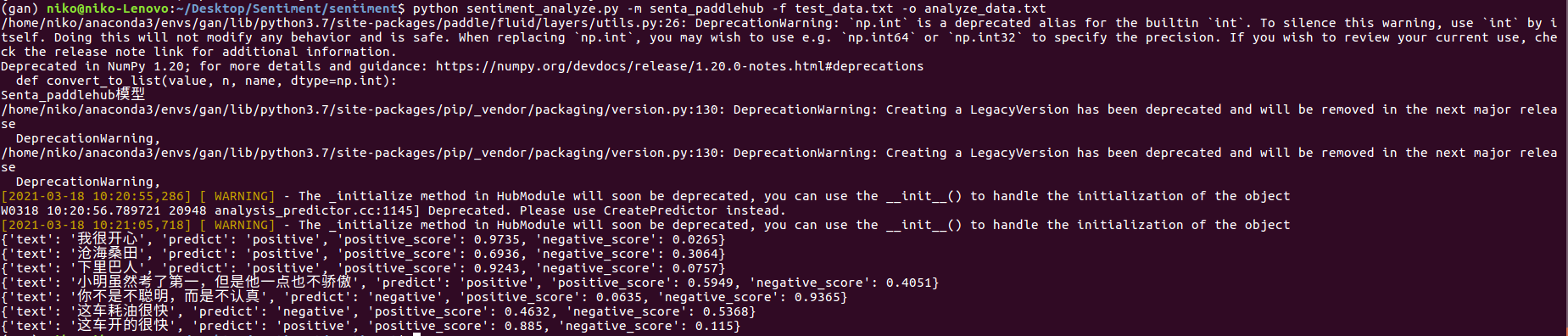
(2) 例：



(3) 脚本测试：

执行命令：python sentiment\_analyze.py -m senta\_paddlehub -f test\_data.txt -o analyze\_data.txt

输出结果：



(5) 优缺点：

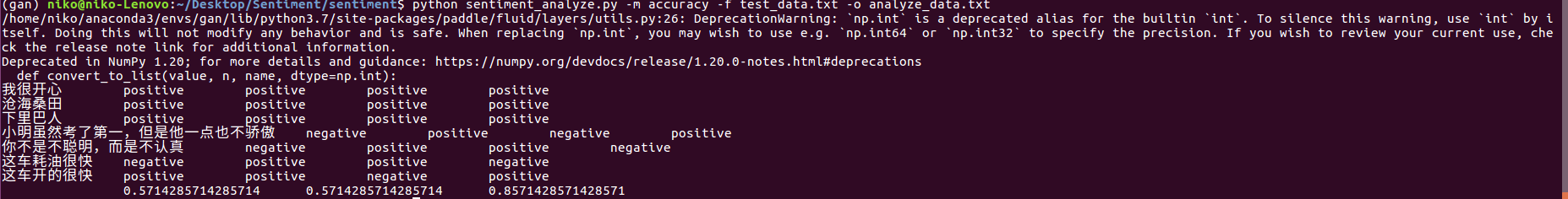
准确率高，不同类数据准确率有高有低，需要测试其他不同方法；只区别积极、消极。

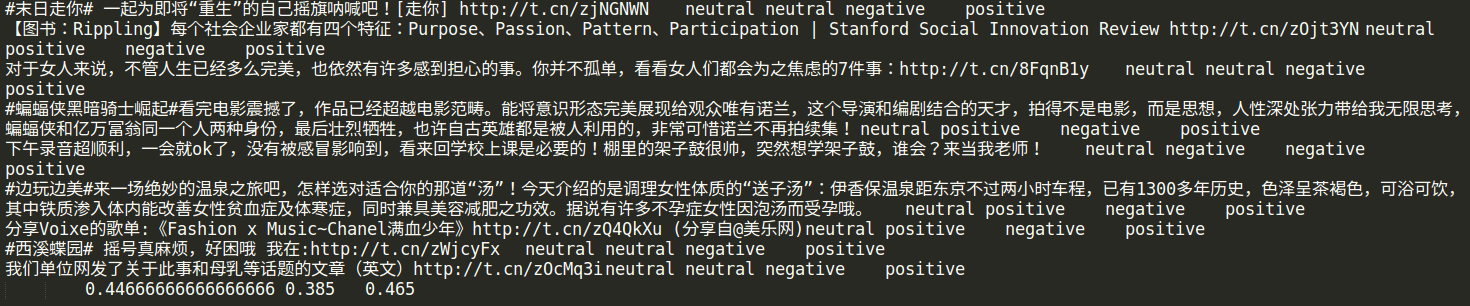
# 4 dict、jiagu、senta\_paddle方法对比

(1) 执行命令：python sentiment\_analyze.py -m accuracy -f test\_data.txt -o analyze\_data.txt

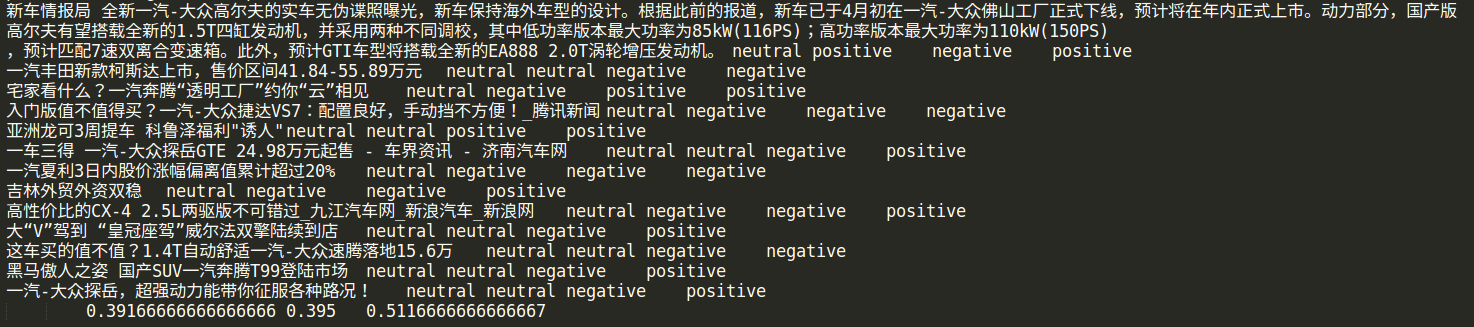
(2) 输出结果：

测试一：

测试二：微博数据积极消极中性个200条



测试三：标题数据积极消极中性个200条



(3) 总结：

准确率：senta>jiagu>dict(dict需要调整scores界限)

速度：jiagu>senta>dict

# 5 情感分析应用场景

### 5.1 华为云调用api

1. 基础版

输入：



输出：



1. 领域版

输入：



### 5.2未来杯“战役”特别赛事-心理疏导线索

<https://ai.futurelab.tv/contest_detail/16#contest_index>

2020年新型冠状病毒肆虐全球，除了身体健康，心理健康同样是这场战役里不容忽视的层面。随着AI的技术的发展，NLP技术应用场景日益广泛，为网络评论文本标注情绪标签，发现潜在的心理疏导需求就是这样一个场景。

在本赛道中，我们采集了近期微博热点事件相关的数万条评论信息作为比赛数据集。参赛选手基于已经标注负面情绪标签的训练集 建立并训练模型，使用该模型为测试集数据标注情绪标签。

任务目标：

为给定的微博评论文本标注负面情绪标签。标签包括：

愤怒、焦虑、无助、悲伤、惧怕、恐慌。

训练数据：

训练数据为来自近期微博热点事件的评论文本及人工标注的负面情绪标签数据。

每条文本关联0至3个负面情绪标签。

开发测试集：

开发测试集用于选手用于自测和模拟提交。

### 5.3 2020语言与智能技术竞赛：事件抽取任务--方案分享

<https://aistudio.baidu.com/aistudio/projectdetail/545914>

竞赛行为-胜负：历经4小时51分钟的体力、意志力鏖战，北京时间9月9日上午纳达尔在亚瑟·阿什球场，以7比5、6比3、5比7、4比6和6比4击败赛会5号种子俄罗斯球员梅德韦杰夫，夺得了2019年美国网球公开赛男单冠军。

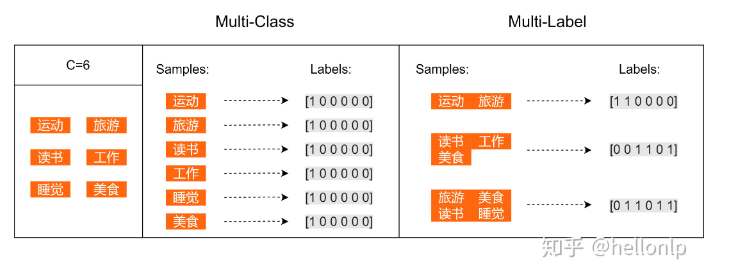
输出结果：



# 6 多标签分类问题

## 6.1 含义

假设个人爱好的集合一共有6个元素：运动、旅游、读书、工作、睡觉、美食。



一般情况下，一个人的爱好有这其中的一个或者多个，那么这就是一个典型的多标签分类任务。

## 6.2 与多分类问题的区别

**·** label\_ids

**·** 交叉熵

**·** 输出概率

### label\_ids

如上图所示

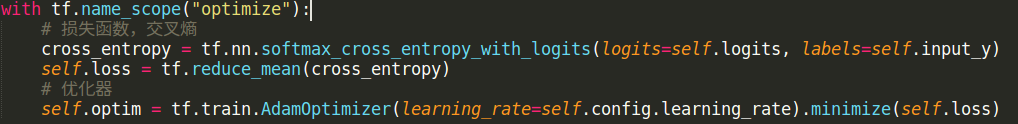
### 交叉熵

在文本分类中，我们使用的交叉熵为tf.nn.softmax\_cross\_entropy\_with\_logits,(0,1)；在多标签文本分类中，我们使用的交叉熵则为tf.nn.sigmoid\_cross\_entropy\_with\_logits,(-1,1)。这样做的原因：

* tf.nn.sigmoid\_cross\_entropy\_with\_logits测量离散分类任务中的概率误差，其中每个类是独立的而不是互斥的。这适用于多标签分类问题。

IMG_256

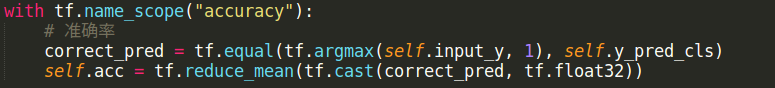
* tf.nn.softmax\_cross\_entropy\_with\_logits测量离散分类任务中的概率误差，其中类之间是互斥的（每个条目恰好在一个类中）。这适用多分类问题。



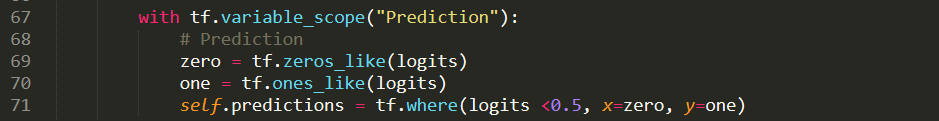
### 输出概率

在文本分类（多元文本分类）中，label\_ids的维度为(batch\_size)；在多标签文本分类中，它的维度为(batch\_size,num\_labels)。这样做的原因：

在多元文本分类中，最后得到的标签只有一个，并且必须是其中的一个。



* 在多标签文本分类中，最后得到的标签可能有1个或者多个。



一般的多元分类是通过tf.argmax(logits)实现，返回的是最大的那个数值所在的label\_id，因为logits对应每一个label\_id都有一个概率。但是，在多标签分类中，我们需要得到的是每一个标签是否可以作为输出标签，所以每一个标签可以作为输出标签的概率都会量化为一个0到1之间的值。所以当某一个标签对应输出概率小于0.5时，我们认为它不能作为当前句子的输出标签；反之，如果大于等于0.5，那么它代表了当前句子的输出标签之一。

## 6.3 实例

<https://github.com/percent4/multi-label-classification-4-event-type>

## 6.4 相关文档

<https://zhuanlan.zhihu.com/p/158622992>

https://cloud.tencent.com/developer/article/1049847

# 附录

## 情感语料地址

1、毕业设计项目,微博语料情感分析，文本分类：

https://github.com/Zephery/weiboanalysis

2、微博数据爬虫：

https://github.com/dataabc/weiboSpider

https://github.com/nghuyong/WeiboSpider

3、为中文自然语言处理领域发展贡献语料

https://github.com/brightmart/nlp\_chinese\_corpus

4、公司名/机构名语料库（Company-Names-Corpus）

https://github.com/wainshine/Company-Names-Corpus

## 相关博客

https://blog.csdn.net/u013510838/article/details/82558797?utm\_term=%E6%83%85%E6%84%9F%E5%88%86%E6%9E%90&utm\_medium=distribute.pc\_aggpage\_search\_result.none-task-blog-2~all~sobaiduweb~default-0-82558797&spm=3001.4430

https://blog.csdn.net/Eudoraff/article/details/100217154?utm\_term=%E6%83%85%E6%84%9F%E5%88%86%E6%9E%90&utm\_medium=distribute.pc\_aggpage\_search\_result.none-task-blog-2~all~sobaiduweb~default-2-100217154&spm=3001.4430