操作流程，按照步骤一步步来，可能会有没有的文件，可能没有放好，但是代码都是可以运行的，可以根据代码内容运行

可能需要安装相应的库，这里的TensorFlow的版本是1.+的

**1. 建立新闻事件分类器**

train使用到了data\_helper代码，首先跑通data\_helper这个代码

1.用data\_sample.zip跑通train.py

输出：trained\_results和checkpoints两个文件夹

训练结果/得到的模型

将输出保存到了训练结果的文件夹中

**2. 针对个股得到事件分类结果**

**2.1fenci.py(在新闻事件分类器文件夹中)**

**运行fenci程序，输入新闻数据集，然后分词出分词后的文件，接着将这个分词后的文件放入getCategory.py**

**2.2**

用xxxx新闻分词.csv跑通getCategory.py

输入：temp\_file.csv

jieba.load\_userdict("./stock\_dict.txt")

stop\_words=pd.read\_csv('./stopwords2.txt', lineterminator="\n", encoding='utf-8',header=None,error\_bad\_lines=False)

title\_file='./中兴新闻分词.csv'

train\_dir="./训练结果/trained\_results\_1594750510/"

输出：predicted\_results文件夹，里面会有分类结果的csv文件

文件里的内容都放在里一个格子里，一个格子内的内容是这样的PREDICTED|DESCRIPT|Date|Bid，后面的代码会处理这个数据

**3.建立新闻矩阵**

newsMatrix.py

输入：分类结果csv文件（predictions\_all\_...csv文件）

输出：中兴新闻矩阵（应该要手动把日期那一列粘过去）

with open ('./中兴新闻矩阵\_3.csv','a+',newline='', encoding = 'utf-8-sig')as cf:

**4.建立财务矩阵：**

1. 删0：手动将原始财务数据中含有0的数据行删区 （先进行没有删除0的）

2. 通过涨跌幅打label.py：

数据集里的中兴通讯10-20财务数据.xlsx

输入：删0后的财务数据

输出：根据涨跌幅判定是涨还是跌并打label，涨标记为1，跌标记为0，生成阈值分别为0.01至0.09的九个csv文件

中兴通讯10-20财务数据\_label001-009.csv

**5. 归一化处理zscore.py：**

中兴通讯10-20财务数据\_label001-009.csv

输入：打label后的csv文件

输出：对数据部分进行zscore归一化处理后的csv文件

中兴通讯10-20财务数据\_label0.01\_zscore.csv

**6.财务和新闻矩阵合并.py：（在原来的代码中输入的是没有归一化处理后的财务矩阵）**

中兴通讯10-20财务数据\_label0.01\_zscore.csv（财务矩阵）

/中兴新闻矩阵\_3.csv （新闻矩阵）

输入：归一化处理后的财务矩阵；新闻矩阵

输出：具有相同日期的财务矩阵和新闻矩阵

Zhongxing财务\_删0\_label0.01\_forVisual\_afterMerge.csv

备注：需要在excel中打开文件，手动调整两个矩阵中日期一列的文本格式（没有日期则需要手动添加）；统一格式才能正确输出。

**7. 合并情感维度：**

**7.1. Bi-LSTM：（使用Bi-LSTM模型训练得到情感预测模型）**

a. ./data/preProcess/fenci.py：

输入：已标注用于训练的新闻数据；newsMatrix.py生成的新闻矩阵

输出：分词后的用于训练的新闻数据；分词后的新闻矩阵

TechNews\_modified\_500\_fenci.csv xdataChina\_modified.csv

b. 通过word2vec得到用于训练的新闻数据的词向量bin文件

c. ./code/Bi-LSTM.py： （这个可能报错）

bi-lstm的原来的文件的测试（中文训练）.ipynb （使用这个）

输入：

- 已进行标注和分词的新闻数据；

- 分词后的新闻矩阵（用于进行预测）

- stopwords.txt

- word2vec.bin

输出：对分词后的新闻矩阵进行预测的结果

d. ./code/merge.py

输入：预测结果（正向情感标为1，负向情感标为0）；新闻矩阵

输出：附上日期信息的情感矩阵（原预测结果文件中仅有一列预测值，merge.py将新闻矩阵中日期等信息合并进来；并将负向情感标为-1）

**7.2. 新闻和情感矩阵合并.py：**

输入：情感矩阵；新闻矩阵（是和财务矩阵合并后的新闻矩阵）

输出：合并后的情感矩阵（保留了与新闻矩阵、财务矩阵一样具有相同日期的数据项）

**7.3. MergeSentimentMatrix.py：**

输入：合并后的情感矩阵

输出：SentiMerge情感矩阵（输入文件中存在大量日期相同的数据项，MergeSentimentMatrix.py将这些同一天的情感值合并在一起，输出一个值代表这一天的情感走向）

**8.models：**

**8.1. finance\_LSTM.py：**

输入：合并后的财务矩阵

**8.2. multi\_input\_LSTM\_event.py:**

输入：合并归并日期后的财务矩阵和新闻矩阵

**8.3. multi\_input\_LSTM\_event+senti.py:**

输入：合并归并日期后的财务矩阵、新闻矩阵和情感矩阵；需要根据前面的文件和程序，准备一个进行了删0和日期合并但未归一化的财务数据文件

输出：预测结果

**8.4. gbdt.py:**

输入：合并归并日期后的财务矩阵、新闻矩阵和情感矩阵

**9.可视化：**

**9.1. 建立可视化输入数据.py：**

输入：multi\_input\_LSTM\_event+senti.py输出的预测结果

输出：满足可视化.py文件所需的csv文件

**9.2. 可视化.py：**

输出：可视化结果