**代码说明**

代码一共包括四部分，分别在layers、models、utils文件夹内，和train.py文件，四部分功能如下：

1. layers： 神经网络层，包含attention、rnn 。
2. models：各模型实现代码，包含ram、cnn 、lstm 等。
3. utils： 对输入数据处理，包括词向量表征文本数据。
4. train.py: 深度学习的主函数
5. **环境配置**

Python运行环境：python 3.6

需要安装的库：pytorch 0.4.1、pandas、numpy、jieba0.39

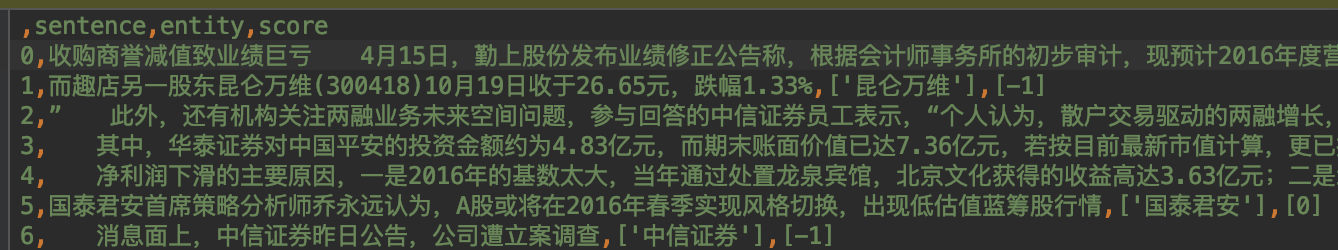
1. **具体使用方法**

* 模型训练：

1. 输入

程序默认的输入文件是data文件下的temp.csv文件。若要更改输入文件，在train.py代码26和28行更改文件路径。

输入文档的格式为utf-8编码的csv文件，每一列内容分别是句子索引、句子内容、句子实体、句子标注结果。样例如下：



其中data文件夹下保存输入数据，temp.csv为演示数据，是单、多实体混合数据集；single.csv/multi.csv为情感词典预测的数据集；signle\_test.csv为8019句标注单实体金融语料数据集;multi\_test.csv为3736句标注多实体金融预料数据集;entity.txt为3000余家上市公司名。

word2vec文件夹内的sgns.financial.word为预训练好的词向量，sgns.financial.char为预训练好的字向量。

1. 模型选择

当前已实现的模型有lstm、at\_lstm、atae\_lstm、cnn、ian、memnet、ram、cabasc、lrg可供选择，经过实验，其结果是单实体ram准确率最高，多实体cnn的准确率最高，单多实体混合，cnn的准确率最高。

代码默认为ram模型，若要更改模型可在train.py第145行更改模型。

1. 字\词向量选择

代码默认为词向量，若要选择字向量可在train.py第146行’word’改为’char’。

1. 运行步骤
2. 打开train.py，在26，28行设置输入文件路径；145行设置模型；146行设置字\词向量，默认参数，输入文件路径/data/temp.csv,深度学习模型ram，选用词向量表征文本；
3. 运行train.py进行模型的训练
4. best文件夹下为训练好的模型参数。
5. 输出结果

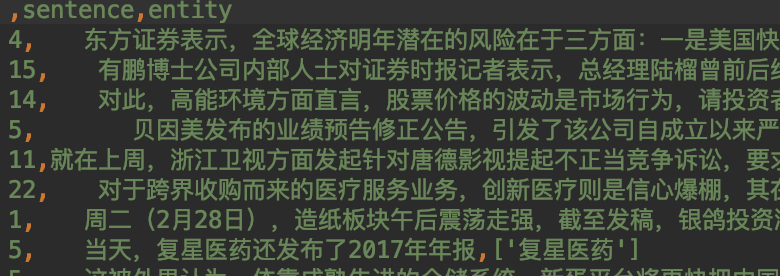
模型的输出结果将被打印在运行栏中，若模型训练后测试集准确率大于79%，该模型训练结果将保存至best文件夹下，当前文件夹下存有ram，cnn，atae\_lstm模型最佳结果； 在当前目录下存有测试集中预测错误的句子，以当前模型名\_error.csv命名，如cnn\_error.csv。

* 预测：

1. 输入：

程序默认的输入文件是data文件下的single\_2500\_temp.csv文件。若要更改输入文件，在train.py代码25行更改文件路径。

输入文档的格式为utf-8编码的csv文件，每一列内容分别是句子索引、句子内容、句子实体。样例如下：



1. 模型选择：

当前已实现的模型有lstm、at\_lstm、atae\_lstm、cnn、ian、memnet、ram、cabasc、lrg可供选择，已训练好的模型有atae\_lstm、cnn、ram，其中ram效果最好，程序默认为ram模型，若要更改模型，在代码59行更改。

1. 字\词向量选择:

代码默认为词向量，若要选择字向量可在train.py第60行’word’改为’char’。

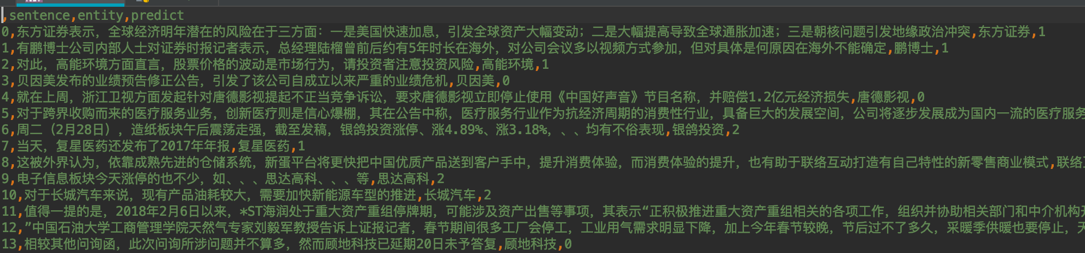
1. 运行步骤：

打开test.py，直接运行即可

1. 输出结果：

在output文件夹下保存模型预测结果，文件名为result.csv

样例如下：

格式为索引，句子内容，公司实体，预测极性

**三．重要代码说明**

主目录下的train.py为主函数，用于初始化模型，以及模型的选择，训练，测试，输出结果。

utils目录下：

dataset.py:处理输入数据，使字\词向量表征输入文本。

word\_embedding.py:加载预训练的字词向量

models目录下：

at\_lstm.py: ATLSTM模型实现代码

atae\_lstm.py: ATAELSTM模型实现代码

cabasc.py: CABASC模型实现代码[1]

cnn.py: CNN模型实现代码

ian.py: IAN模型实现代码[4]

lrg.py: LRG模型实现代码

lstm.py: LSTM模型实现代码

memnet.py: memnet模型实现代码[3]

ram.py: RAM模型实现代码[2]

layers目录下：

attention.py:实现注意力机制层功能

dynamic\_rnn.py:实现基础RNN层功能

data目录下：

word2vec文件夹：sgns.financial.word预训练的词向量；

sgns.financial.char预训练的字向量

entity.txt:3000余家上市公司名

single.csv：从65535篇金融语料中处理的单实体数据集（情感词典预测结果）

multi.csv:从65535篇金融语料中处理的多实体数据集（情感词典预测结果）

single\_test.csv:已标注的8019句单实体

multi\_test.csv:已标注的3037句多实体

temp.csv:8019句单实体和3037句多实体混合

best目录下：

atae\_lstm\_besk.pkl:训练好的atae\_lstm模型参数

cnn\_besk.pkl:训练好的cnn模型参数

ram\_besk.pkl:训练好的ram\_lstm模型参数

**四．常见问题**

1. 深度学习代码默认是GPU加速，所以若电脑未装GPU可能会报错。
2. 程序运行需要一定的内存，最好在内存较空闲的时候运行。
3. 运行环境建议在安装anaconda3后运行。

模型参考以下论文：

[1] Liu, Qiao, et al. "Content Attention Model for Aspect Based Sentiment Analysis." Proceedings of the 2018 World Wide Web Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2018.

[2] Chen, Peng, et al. "Recurrent Attention Network on Memory for Aspect Sentiment Analysis." Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017.

[3] Tang, Duyu, B. Qin, and T. Liu. "Aspect Level Sentiment Classification with Deep Memory Network." Conference on Empirical Methods in Natural Language Processing 2016:214-224

[4] Ma, Dehong, et al. "Interactive Attention Networks for Aspect-Level Sentiment Classification." arXiv preprint arXiv:1709.00893 (2017)