PRML Assignment-4 报告

16307130076 赵伟丞

对于FastNLP的看法和建议

之所以将这个部分放在第一部分，是因为我在这个Assignment中并没有使用fastNLP框架。事实上，我原本是有尝试的（包括在Assignment 3中也有使用），但是经过了一定程度的使用后，我不认为这不能比直接使用Pytorch框架更能帮助我解决实际问题，原因主要有：

- 一些功能上的缺陷和缺失
- 在本人做这个Assignment的关键时刻出现的更新的乌龙事件
- 实际上写的代码与直接使用PyTorch没有什么区别（和第一点有一点重合，也是对于我来说最为主要的一点）

尽管我没有使用fastNLP框架，但是我仍然认为其在未来会是一个非常实用的自然语言处理框架，特别是如果一个很好的对于PyTorch相关模块的封装得以实现的话，这将对于对PyTorch或者深度学习了解不深的其他专业领域（如语言研究的学者）利用深度学习的方法进行自然语言的研究是非常有帮助的。以下是我认为的亟待改进的地方：

- fastNLP中的Vocabulary，在min_freq参数不为0的情况下，对于同样的一份数据集，转化为字典的结果每次的执行结果会有不同，产生的词典长度会有个位数的差异。然而Vocabulary并没有提供一个保存与读取框架，这就导致除非将Vocabulary对象中的word2idx和idx2word手动保存下来（本人在Assignment 3中使用json的方式将这两个字典保存了下来），甚至没有办法让train和demo中使用同样的Vocabulary来运行已有模型，这是一个急需解决的问题。
- fastNLP目前缺少一个分句的功能，同时分词也是通过python的spilt来让用户手动实现，同时也缺乏一个停用词（比如英语中的“the”，“is”，“are”）去除的功能。这些在fastNLP的依赖nltk中都有实现，fastNLP完全可以提供一个相应的封装来为用户直接提供这个必要的功能。特别对于长文本的处理，一个树形结构（文本-句子-词语-语素）是很显然的，如果简单地将一整段文本拆成一个一个词单独处理，我不认为这将对于文本的处理有很大的帮助。因此，上述的功能十分有必要。
- 同样的我建议fastNLP类似sklearn.dataset一样提供直接加载数据集和预训练字典的功能，因为若是想要实现一个对于自然语言处理方案的较好封装的话，我认为这些必须的材料如果还需要用户手动载入，手动预处理未免有些不方便，相比之下如果一个很好的封装得到了实现，这将会非常有助于我在上文提到的非计算机相关从业人员使用这些框架。
- 另外，目前很多我认为很有必要的功能还是需要用户自己手动实现，如2维的padding等，当然这个可以逐步地去实现。

总而言之，我相信fastNLP是一个非常有未来的框架，将会非常有助于非计算机专业的人士进行交叉研究，而作为研究的角度，我认为直接使用PyTorch是一个跟你国家不错的选择。因为全新的研究需要大量自定义实现的方法，这个时候使用框架与不使用实在没有什么太大分别。

额外数据集

本人在这个Assignment中使用了以下数据集：

数据集名	类别数量	训练集大小	测试集大小

数据集名	类别数量	训练集大小	测试集大小
20Newsgroup	20	11 314	7 532
AG’s News	4	120 000	7 600
DBPedia	14	560 000	70 000

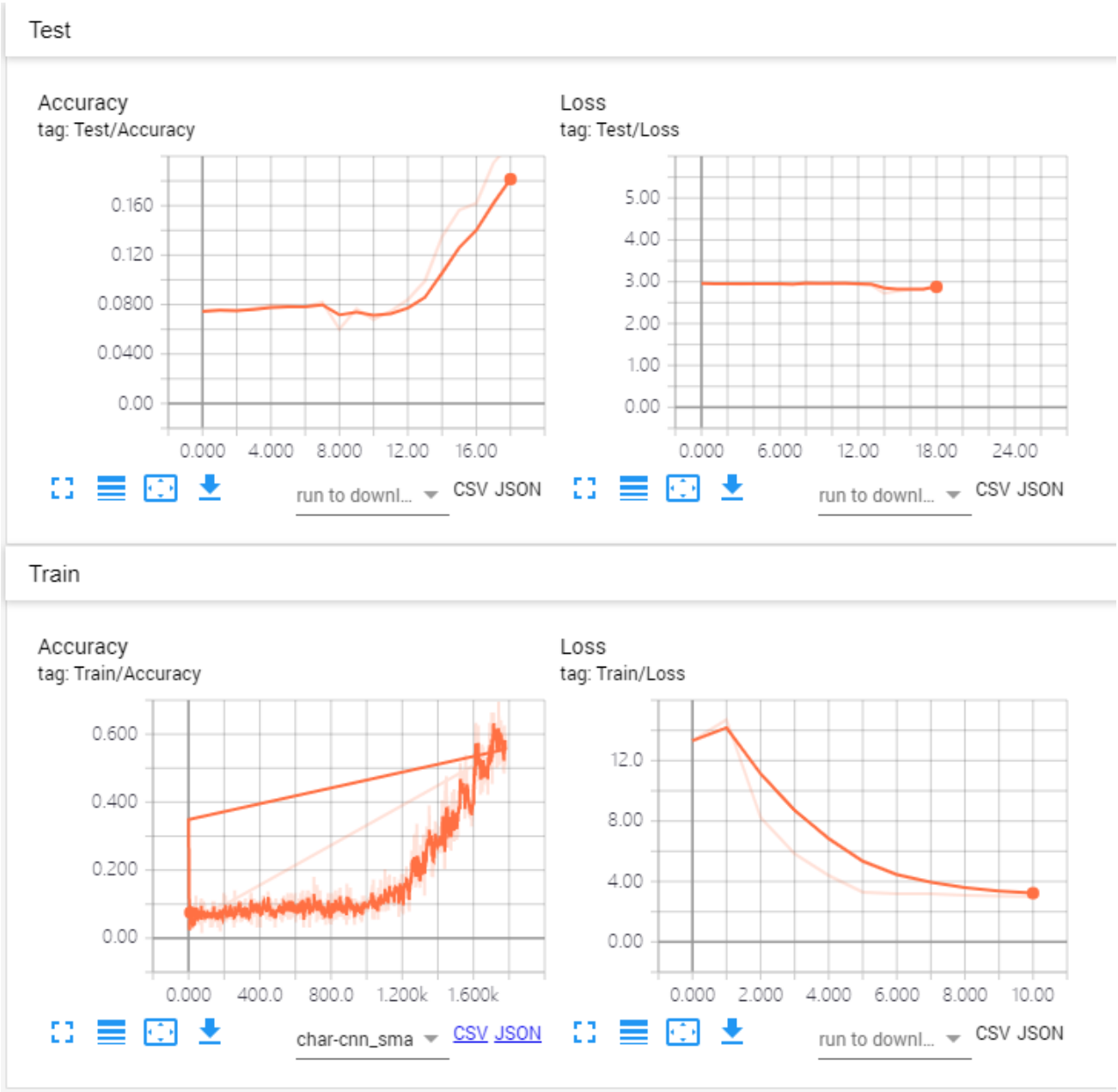
Character Level CNN

Character Level CNN 采用一个包含7层2维卷积和3层全连接层的卷积神经网络，在字符级别对文本进行分类。对于每个文本，截取或补齐至1014个字符，转化为小写，并且以"""abcdefghijklmnopqrstuvwxyz0123456789,;.!?:'"/\|_@#$%^&*~`+-=<>()[]{}"""作为字典进行one-hot编码，形成一个字典大小*序列长度的输入，以分类类别作为输出。优化器采用Adam，学习率选择1e-3，batch_size为128，采用交叉熵作为损失。（由于出现了一些情况，early stop没有启用）

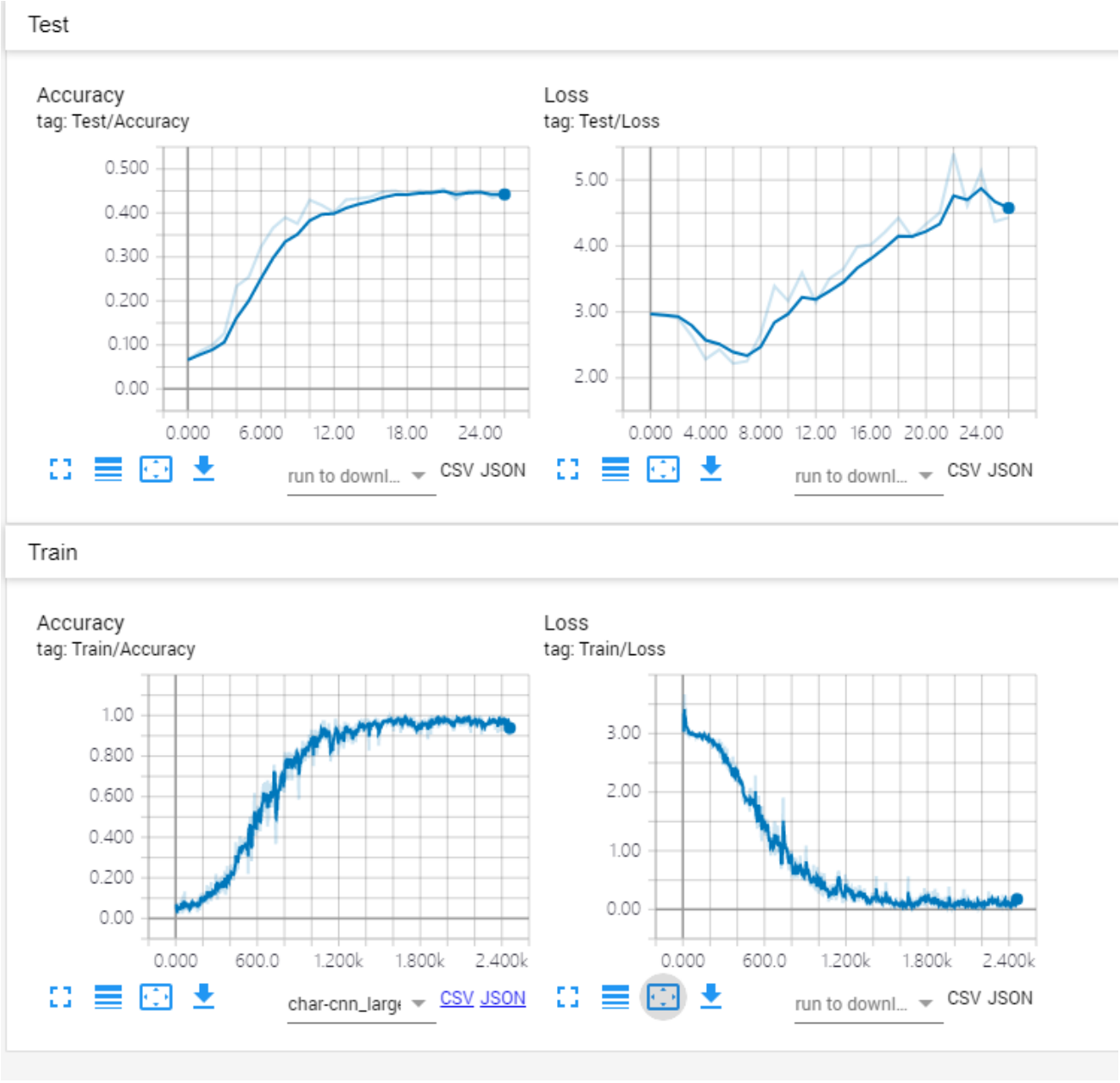
以下为结果：

Character Level CNN 包含large和small两种feature方式，我首先在20newsgroups上分别做了测试，发现small的效果不尽如人意，因此选择了large做了两个额外数据集的测试。

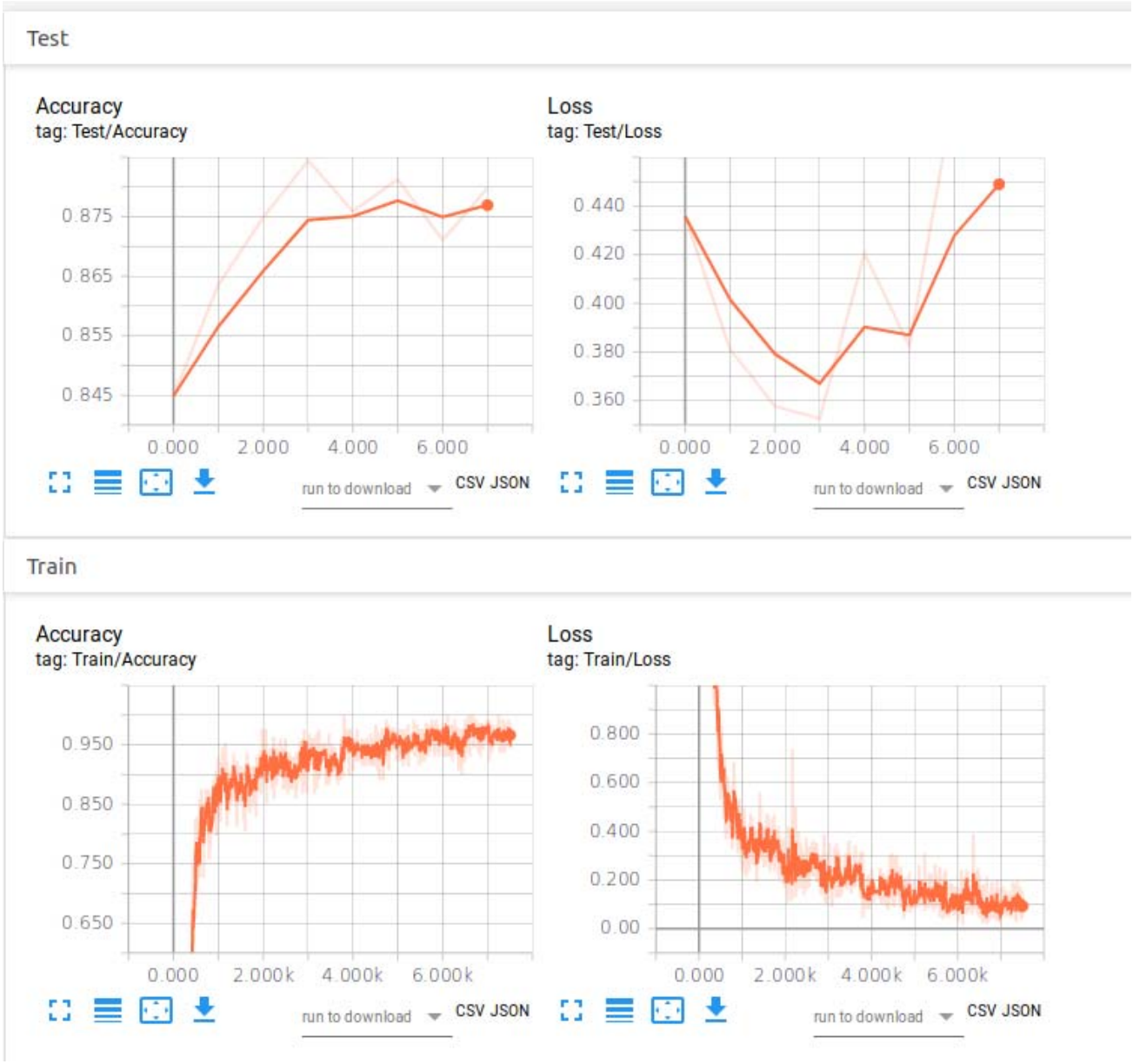
20 news group (small feature)



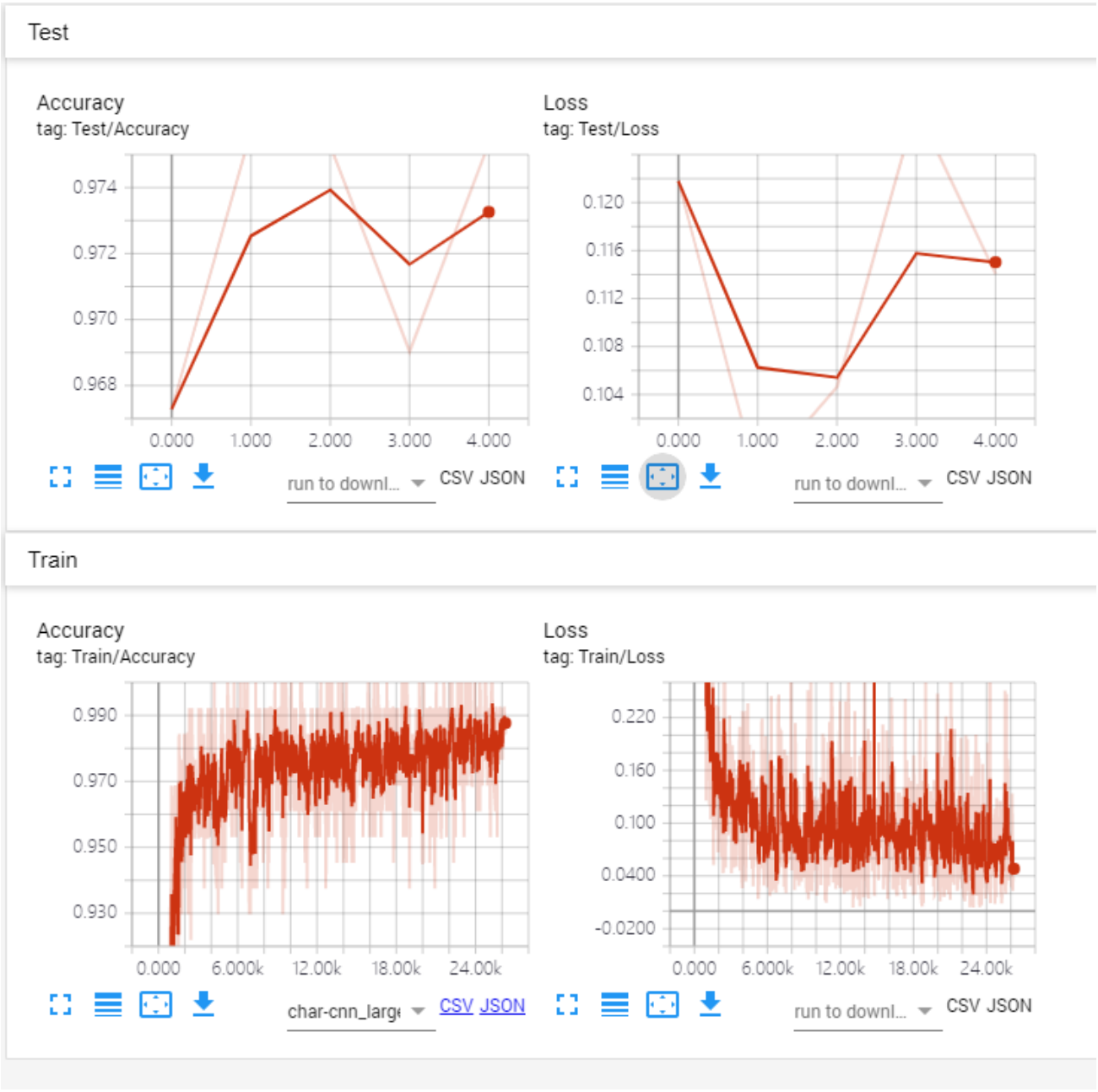
20 news group (large feature)



agnews (large feature)



dbpedia (large feature)



因此最终结果为（large）

数据集名	类别数量	训练集准确率	测试集准确率
20Newsgroup	20	96.88%	45.40%
AG's News	4	96.09%	87.70%
DBPedia	14	99.02%	97.31%

思考与疑惑：尽管我已经在最后的全连接层部分加上了0.5的dropout，但是在20newsgroup数据集上出现了一定程度的过拟合现象，导致测试集准确率不高。除此之外在这个数据集上还出现了训练集Loss与准确率同时上升的情况（这也让我没有在进行这个任务的时候启用early stop），尽管这个情况可以在一定程度上通过dropout层在测试时并没有启用来解释，但是我认为这可能不是全部原因，不知助教对此有没有什么高见?

Hierarchical attention networks

Hierarchical attention networks 采用一个具有Attention的树形网络来完成文本分类。对于每个句子，采用一个双向的GRU进行计算，基于GRU的每一个输出，计算一个注意力系数，对整个句子的输出加权求和后作为这个句子的向量表示。同样的，对于文本中的每一个句子，将上述过程计算出的句子的向量表示也经过一个双向GRU然后类似的计算一个注意力矩阵，类似地进行加权求和，如此下来再将最后文本的向量表示通过线性变换来映射到最后的分类输出。

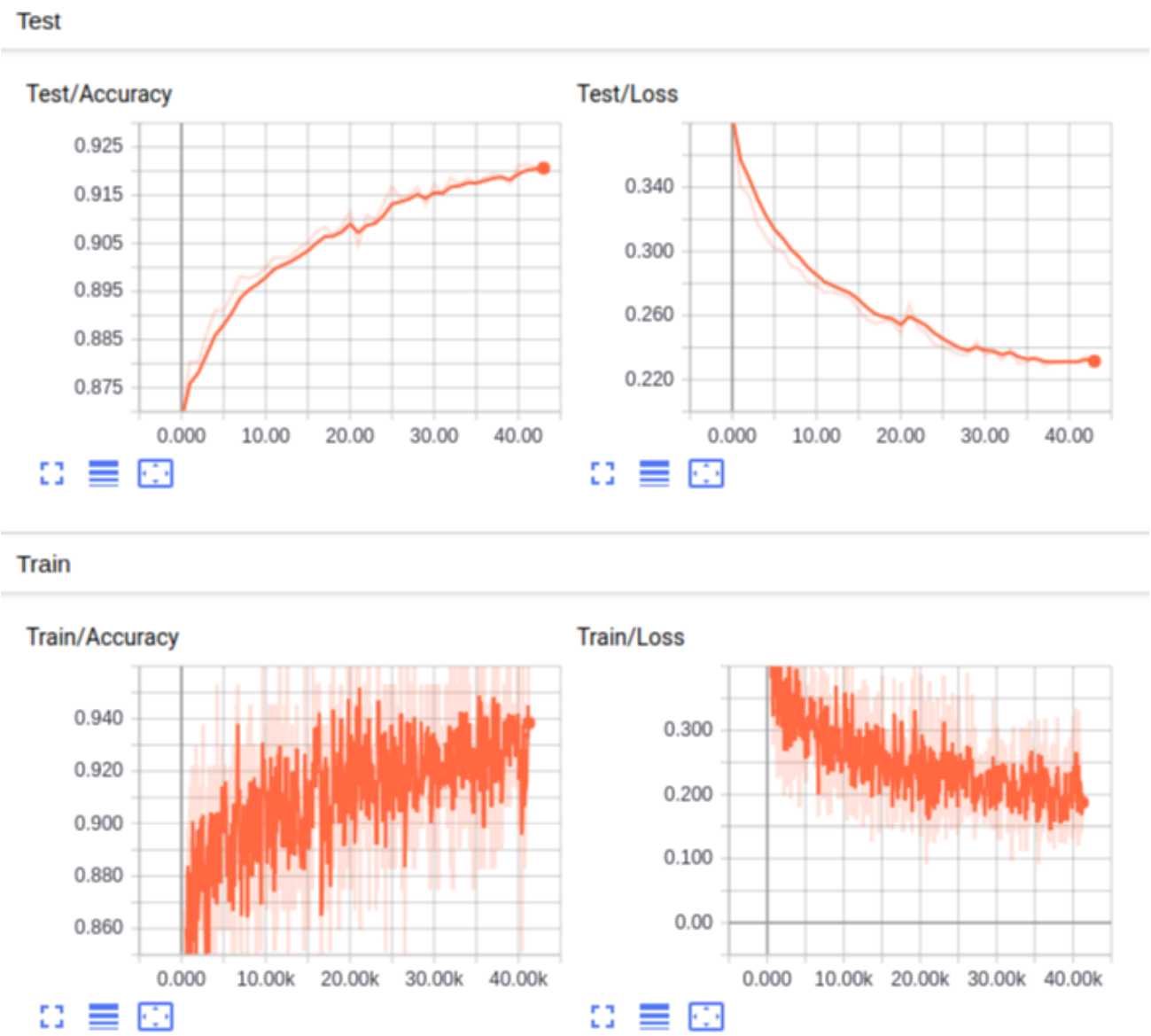
网络采用nltk对文本进行分句和分词，然后使用glove.6B.50维预训练词向量对embedding进行初始化，隐藏层大小相应的也是采用50，每句句子截断或补齐至35词，文本截断或补齐至30句作为输入。优化器采用SGD，学习率设为0.1，momentum设为0.9，batch_size为128，采用交叉熵作为损失，3个epoch没有进步则early stop。

以下为结果：

20 news group

由于这个数据集数据较脏，无法完成分句工作，导致无法进行训练。

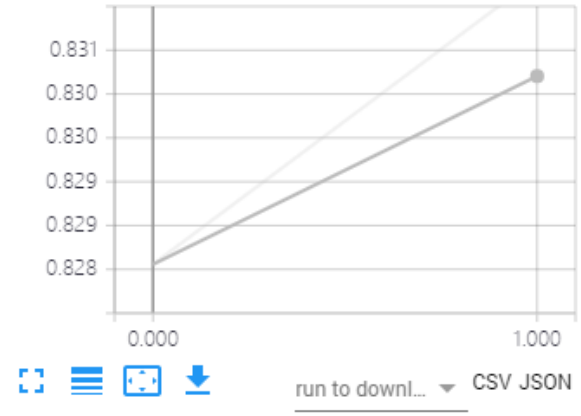
agnews



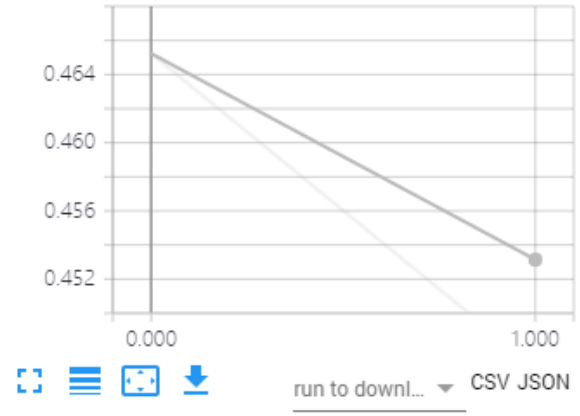
dbpedia（由于时间所限，训练不够充分，只训练了一个epoch）

Test

Accuracy
tag: Test/Accuracy

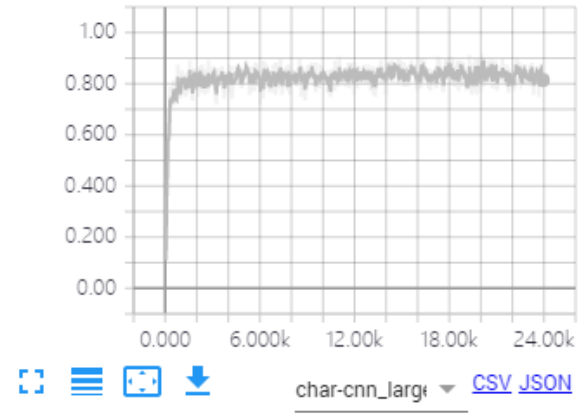


Loss
tag: Test/Loss

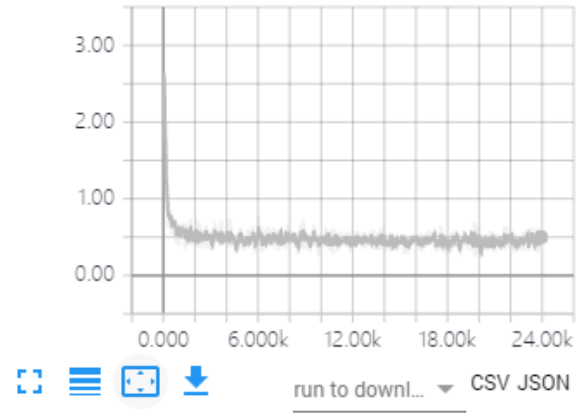


Train

Accuracy
tag: Train/Accuracy



Loss
tag: Train/Loss



更新：dbpedia

因此最终结果为

数据集名	类别数量	训练集准确率	测试集准确率
20Newsgroup	20	N/A	N/A
AG's News	4	93.69%	92.16%
DBPedia	14	83.05%	85.65%
DBPedia (充分训练)	14	99.23%	98.65%

关于这个算法的思考：由于这个算法需要文本呈现出文本-句子-词语的树形结构，如果文本无法被区分出这个树形结构，就很难进行很好的分类。同样的，由于有Attention的存在，需要训练的权重比较多，因此需要选择一个较大的学习率来使得模型尽快收敛，这也一定程度上限制了最后结果的准确性。