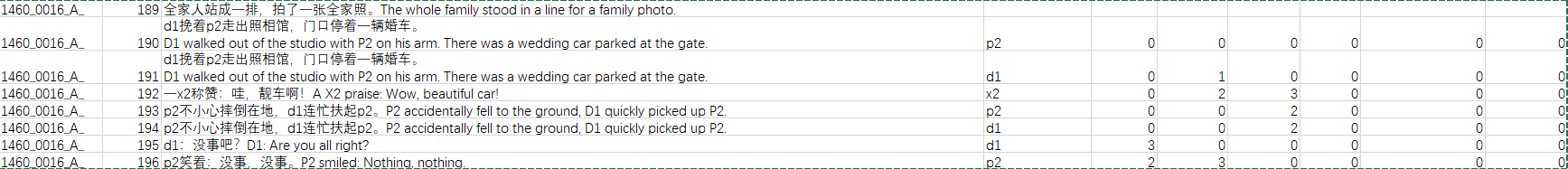
摘要：

This task is to analyze and identify the emotions of each character involved in every dialogue and action description in the script scenes from multiple dimensions. Comparing with traditional sentimental classification task, there are more changes in this task. Emotions are multidimensional, and each emotion has a degree. For example, the degree of happiness ranges from 0 to 5, with 0 being none and 5 being the strongest. A sentence may have a variety of emotions, such as joy, surprise. Emotion classification is for a certain role in a sentence, rather than the whole sentence. A sentence may have multiple roles with different emotions. Considering the property of the task, we tried a few networks which different from what the multi-classifier does.

数据：（做成表格，而不是图片，第一个情绪改成‘爱love’）





The above table is an example: The table content contains the script of a movie. The character column contains the specified character, that is mentioned in the script. The last 6 columns are the labels, which is in the training data but missing in the test data. The task is to identify the given character’s six emotions: love, happiness, surprise, anger, fear, and sorrow, and numerically rank them according to the script . A sentence has multiple characters, such as p2, d1 and x2, and for each character, the type and degree of emotion needs to be identified. In the sample, there is one line: A X2 Praise: “Wow, beautiful car!", which contains two emotions: "joy" and "surprise", and they are in degree 2 and 3, respectively.

带标签的数据总共有36819条，我们随机打乱后按8:2的比例分割成训练集和验证集。我们统计了在训练集上label分布，从数据分布中明显看出情绪值为0的占绝大多数。情绪值越大，占比越小。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| degree | love | joy | surprise | anger | fear | sorrow |
| 0' | 28434 | 27262 | 27735 | 26397 | 27048 | 24898 |
| 1' | 420 | 1645 | 1033 | 1612 | 1253 | 2259 |
| 2' | 328 | 370 | 458 | 981 | 815 | 1594 |
| 3' | 273 | 180 | 229 | 465 | 339 | 753 |
| percentage of 0 | 0.965337 | 0.925485 | 0.941606 | 0.896181 | 0.918282 | 0.843886 |
| percentage of 1 | 0.014259 | 0.055844 | 0.03507 | 0.054728 | 0.042539 | 0.076566 |
| percentage of 2 | 0.011136 | 0.012561 | 0.015549 | 0.033305 | 0.027669 | 0.054027 |
| percentage of 3 | 0.009268 | 0.006111 | 0.007775 | 0.015787 | 0.011509 | 0.025522 |

评估指标：

本任务采用常用的均方根误差（RMSE）来计算评分，按照“文本内容+角色名”识别出的6类情感对应的情感值来统计。

R M S E=\sqrt{\frac{\sum\_{i=1}^{n} \sum\_{j=1}^{6}\left(y\_{i, j}-x\_{i, j}\right)^{2}}{6 n}}*RMSE*=6*n*∑*i*=1*n*​∑*j*=16​(*yi*,*j*​−*xi*,*j*​)2​​

score = 1/(1 + RMSE)

其中是yi,j预测的情感值，xi,j是标注的情感值，n是总的测试样本数。  
**最终按score得分来排名。**

方法一

我们用了最简单的方法实现了baseline版本，得到的分数为测试集0.68,在验证集上0.6787680571339884。方法详见如下所述

1. 把角色名称与对话合并成一条文本，对角色的情绪识别变成对文本分类问题，但由于有多个情绪需要识别，因此是多标签分类问题。每个标签的类别是[0,1,2,3]四种取值，但是从数据分布来看，类别2和3的占比非常少，为了简化，我们采用多标签二分类的方法。
2. 把数据打乱按8:2比例分割成训练集和验证集
3. 多标签分类和多分类的区别在于：
   1. 多标签分类是对一个样本的多个方面，或者多个目标进行分类，通常是二分类。这些方面不是互斥的，比如对于一张人物图片样本，在性别方面区分是男是女，在年龄方面区分是大人还是小孩，在种族方面区分是东亚人还是西方人。图片的这几个方面可以同时出现，既有性别，又有年龄，还有种族
   2. 多分类问题是对一个样本判断属于哪个类别，类别的取值超过2个。这些类别是互斥的，属于类别1就不可能属于类别2或类别3。比如对人脸识别任务是把人脸分类成不同人的脸。
4. 我们的任务是个多标签分类，因为对于一句文本，要从多个不互斥的方面（爱，乐，惊，怒，恐，哀）进行分类，为了简化，我们仅仅分类为0类或1类，而类别2和类别3当成类1来看待。
5. 把文本转成向量，我们采用目前比较流行且效果比较好的bert-base向量
   1. 我们取一批文本样本，批大小为8，算出这批数据的最大文本长度max\_length\_batch,把文本的每个字符先转成字符id，把一批长短不一的文本padding到固定长度max\_length\_batch的序列。Padding值为0
   2. 将样本的字符id序列送入bert模型，从bert模型最后一层的第0个单元取出向量作为整个句子的向量。
6. 把得到的句向量送入全连接的网络进行分类，由于是6个标签，输出是6个神经单元，得到6个logit。
   1. 与多分类不同，多分类任务是把6个logit做softmax，而多标签分类任务是把6个logit一一做sigmoid，每个用sigmoid激活后的值代表这个标签为类别1的概率
   2. 把概率大于0.5的标签，转为1，小于等于0.5的标签转为0
7. 损失函数采用二分类的交叉熵。

方法一的缺点

1. 方法一为了采用现成的工具包simpletransformer,快速实现任务，我们对任务做了简化，把情绪值的识别简化成多标签二分类。而实际上情绪值有0,1,2,3这四个取值，属于多分类，但不同于普通的多分类，这个分类的各个类别是情绪的强度值，其实是有大小之分的，如果当成普通分类问题，采用交叉熵损失，不能反应强度值1与强度值3之间的误差比强度值1到强度值2的误差更大的事实，所以直觉上，我们认为转成回归问题比当成分类问题更合理。
2. 由于情绪识别是针对指定角色的，方法一简单的把角色字符合并到剧本的文本里，从而当成对文本的分类问题。但是角色在这里起到了至关重要的作用，如果合并到文本中，相当于把角色与文本中的其他字符同等对待，这使得模型有机会忽视了角色的重要作用。
3. 通过对数据分析知道，情绪值不仅仅与当前的文本有关，而且与上文情节有关。同样一句话放在不同的上下文里所表现出的情绪是不同的，而方法一没有利用上文信息。
4. 因此，基于以上的分析，我们逐步的做了改进，从而逐步提高了在验证集和测试集上的分数

方法二

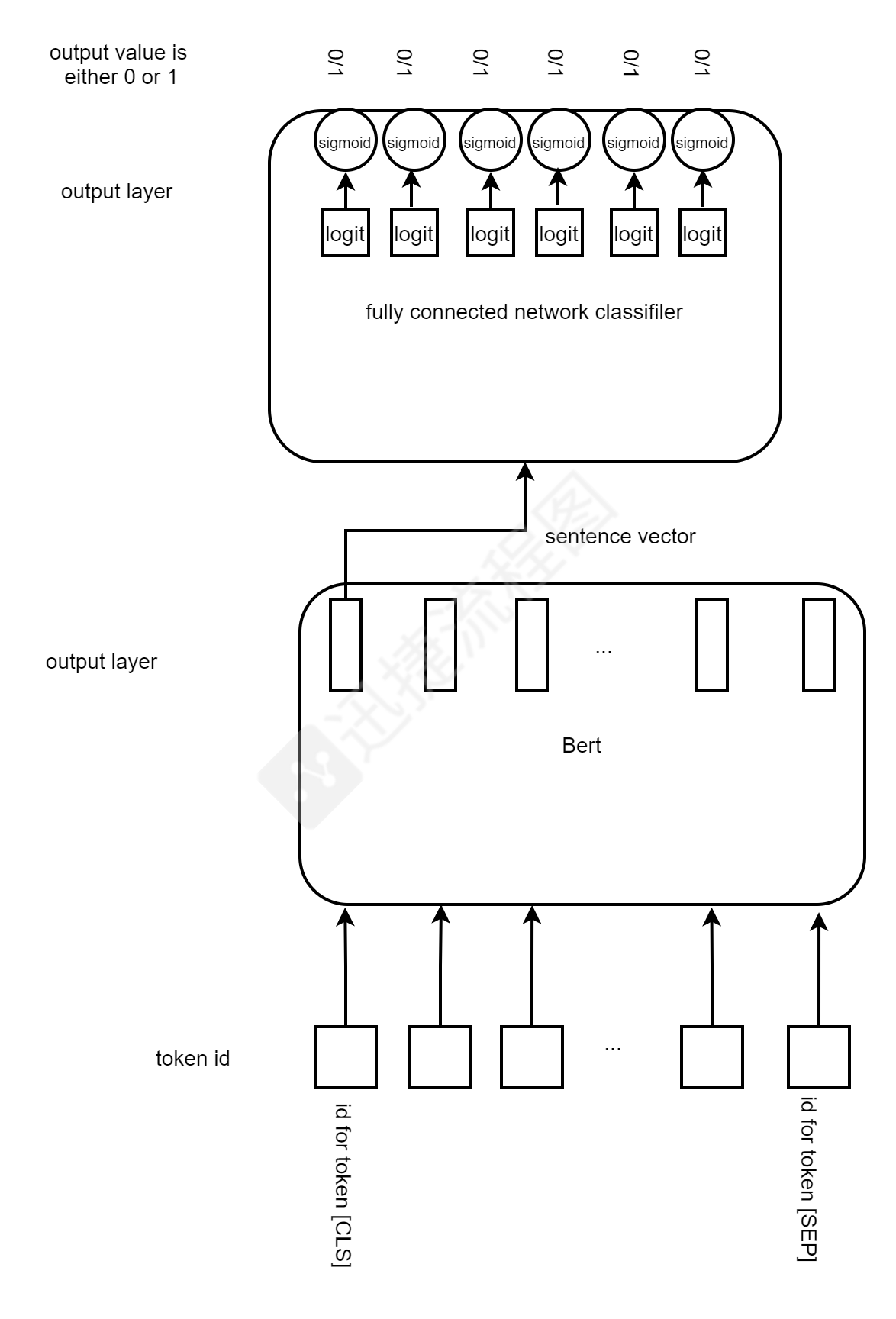
基于线性回归的分类。我们采用多个输出块，一种情绪对应一个输出块，每个输出块内部是2层全连接网络，最后一层输出维度1，不加任何激活函数，直接用logit作为线性回归的输出，损失函数使用MSE。得到的logit是一个实数，我们对logit做四舍五入，从而得到取值范围是[0,1,2,3]整数。6种情绪的情绪值对应6个输出块的整数值。由于有多个输出块，总的损失是所有输出的损失和。这实际上是一种多任务的学习方法。每个任务是对其中一种情绪进行识别，但是共用句向量

方法三

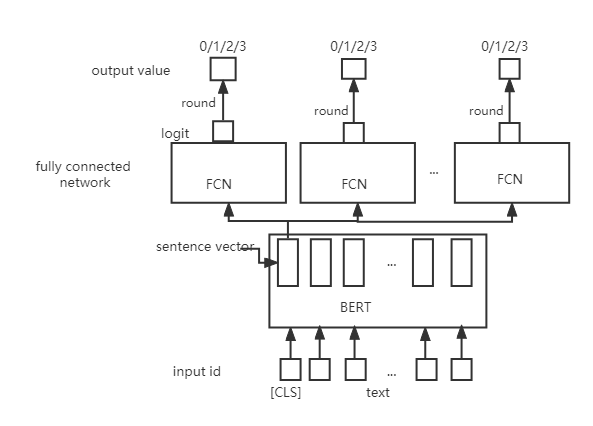
在方法二的基础上，我们仿照BERT模型处理问答任务的方式，把角色字符串当成问答任务中的问题text1，把文本当成问答任务的参考文本text2，但是与问答任务不同的是，问答任务是针对序列的每个字符进行分类，从而得到答案在文本中的起始位置和结束位置。我们这里仍然从输出层的第0个位置取出向量当成这个文本和角色的向量表示，这个向量表示再作为特征输入到输出块进行处理。由于在bert内部，text1和text2实行注意力机制，因此模型对角色的重视程度比方法一高得多。

方法四

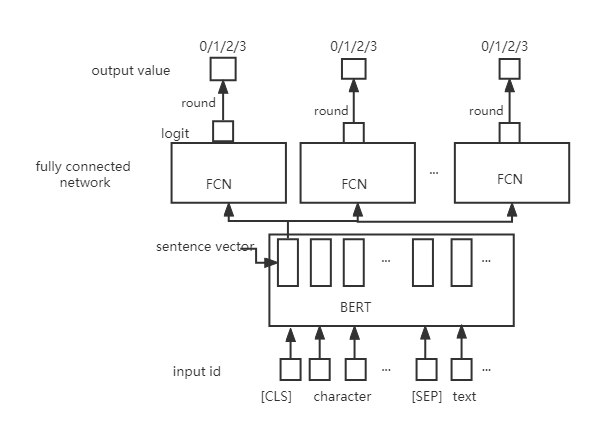
在方法三的基础上，我们引入了历史文本。由于时间有限，我们暂时只引入2句上文，把前面2句文本简单的与当前文本进行合并，得到一段更大的文本。为了实现对历史文本的引用，我们对数据进行了预处理。我们把剧本数据中的content按剧情先后顺序进行排序，并且去重，编上序号content\_id，作为一个剧本库。我们把原训练数据和测试数据的content找到content\_id,然后把content\_id-2,content\_id-1与content\_id所对应的三条文本从剧本库里取出，按先后顺序合并成一个大的文本。这样我们利用了历史文本信息对当前文本的角色情绪进行预测。由于模型的文本长度不能无限增加，所以我们暂时只引用2句上文。未来我们打算增加一层attention层，多个句子分别得到bert句向量后，送到attention层与角色向量进行注意力计算得到针对更长文本的句子表示，再送输出曾经进行情绪值预测。



方法一结构图



方法2结构图

方法三结构图